ACL 2025

The 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)

Proceedings of the Conference - Volume 2: Short Papers

July 27 - August 1, 2025

The ACL organizers gratefully acknowledge the support from the following sponsors.

Diamond

Alibaba Cloud

蚂蚁集团 ANT GROUP

Bloomberg Engineering

CITADEL | CITADEL Securities

cohere

Google Research

LinkedIn

Meta

mi

Platinum

amazon | science

Baidu 百度

ByteDance

HUAWEI

IBM

Megagon Labs

ORACLE

Tencent Toloka

Gold

CHEN TIANQIAO & CHRISSY INSTITUTE KUAISHOU Thomson Reuters™

Silver

Appen Adobe nexdata

Bronze

Babelscape® DataoceanAI DEFINED.AI JPMorganChase

NatWest Group NEC translated.

Diversity and Inclusion

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-252-7

Message from the General Chair

It is my great pleasure and honor to welcome you to the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025), held in beautiful Vienna, Austria, from July 27 to August 1, 2025. ACL 2025 continues our field's tradition of excellence in scholarship, innovation, and inclusivity, and I am deeply grateful to the many volunteers who have worked tirelessly to bring this event to life.

I want to express my deepest thanks to our Program Chairs — **Wanxiang Che**, **Joyce Nabende**, **Mohammad Taher Pilehvar** and **Ekaterina Shutova** — who have overseen the reviewing and selection process, and shaped a compelling and diverse scientific program. Overall, we received more than 8300 submissions and accepted 1700 at the main conference and 1392 as findings. This was made possible through close coordination with the ACL Rolling Review (ARR) team, led by the ARR Editors-in-Chief — **Jun Suzuki**, **Jing Jiang**, and **Xiaodan Zhu** — and I sincerely thank them. Although some improvements are still possible, the ARR workflow was already well integrated with the conference review and decision-making process. We are also deeply grateful to the (many!) Senior Area Chairs, Area Chairs, reviewers, and the Best Paper Committee (led by **Rada Mihalcea** and **Roi Reichart**), whose dedication ensured the high quality of our program. I am also grateful to our Technical Open Review Chairs, **Niket Tandon** and **Lizhen Qu**, for their behind-the-scenes work. For those joining virtually, our Virtual Infrastructure Chairs — **Manling Li**, **Yang Liu**, and **Avi Sil** — have worked to make the hybrid conference experience inclusive and engaging. I would also like to recognize the efforts of our Ethics Chairs — **Karën Fort** and **Bjorn Ross** — who have led the work aimed at ensuring that submissions meet the ethical standards of our field.

ACL 2025 hosts a rich set of 28 workshops and 8 tutorials, thanks to the dedicated efforts of our Workshop Chairs — **Terra Blevins** and **Christophe Gravier** — and our Tutorial Chairs — **Yuki Arase**, **David Jurgens**, and **Fei Xia**. Our Demonstration Track Chairs — **Pushkar Mishra**, **Smaranda Muresan**, and **Tao Yu** — have put together an impressive set of system demonstrations of cutting-edge innovations in NLP. I am proud to say that we also have a rich Industry Track, thanks to **Yunyao Li** and **Georg Rehm** who created a "conference in the conference". Given the growing interaction between academia and industry, I hope this tradition, which stems from NAACL and which I reinstated at ACL, will be continued in the following editions.

As is customary, TACL and CL papers are presented at the conference. Thanks go to the TACL Editors-in-Chief (**Asli Celikyilmaz**, **Roi Reichart**, and **Dilek Hakkani-Tur**) and the CL Editor-in-Chief, **Wei Lu**, for their coordination efforts. I am also proud to announce that this year we are introducing the new CL Doctoral Dissertation Award at the conference. Other notable innovations include virtual presentation sessions held in parallel with the in-person poster sessions, a reception event dedicated to Findings presentations, and the introduction of combined oral and panel sessions designed to foster more thorough and interactive research discussions.

This conference could only have been made possible thanks to the invaluable work of our ACL Business Manager and Director of Events, **Jenn Rachford**, and her team — in particular, **Megan Haddad**, our ACL Administrator and Registration Manager — especially given the largest number of participants ever at ACL, estimated at the time of writing to be around 5000! Thanks also go to **Damira Mršic** and the Underline team for managing all content for the hybrid experience. The Local Organization Chairs, **Benjamin Roth** and **Dagmar Gromann**, worked hard on the bidding process, providing huge support with all local arrangements, the social event, and the visa process, eventually ensuring an outstanding experience for our in-person attendees. Our Visa Chairs — **Rexhina Blloshmi** and **Eleni Ilkou** — have helped numerous participants with their visa needs and any logistical difficulties in attending the conference. I also thank our Student Volunteer Chairs — **Pedro Henrique Luz de Araujo** and **Eleonora Mancini** — for their behind-the-scenes work, which will prove indispensable during the conference.

Additional thanks go to **Zhu Liu**, **Mingyang Wang**, and **Jin Zhao**, our Student Research Workshop Chairs, who, with the support of faculty advisors **Lea Frermann**, **Daniel Hershcovich**, and **Tristan Miller**, devoted their work to supporting and mentoring the next generation of researchers. They also secured additional funding from the Vienna Meeting Fund — congratulations!

Our Publication Chairs — **Pierpaolo Basile**, **Libo Qin**, and **Zhenghao Liu** — have ensured timely and high-quality conference proceedings. I also thank the Handbook Chairs, **Els Lefever** and **Qiongkai Xu**, for assembling the handbook that will guide attendees through the conference program.

Of course our conference could not exist without the support of our sponsors! We are deeply grateful to our Sponsorship Chairs — **Raffaella Bernardi** and **Thomas Scialom** — and to **Chris Callison-Burch**, ACL's Sponsorship Director, for securing the generous support that enables us to keep the conference accessible to as many participants as possible.

Fostering an inclusive environment remains a key goal of ACL. Our Diversity and Inclusion Chairs — **Senja Pollak**, **Maria Ryskina**, **Shane Storks**, and **Hwaran Lee** — have worked to support diverse participation and organize activities that reflect the global nature of our vibrant community. An inclusive ACL allows researchers from all areas and backgrounds to contribute, collaborate, and thrive — enriching both the scientific conversation and the broader societal impact of our work. We remain committed to building a community where everyone feels welcomed and valued.

Thank you to the Publicity and Social Media Chairs — **Anette Frank**, **Shruti Rijhwani**, and **Horacio Saggion** — for their communication work through social media, extending the conference's reach to a wider audience. And this brings me to our Website and App Chairs — **Xudong Han** and **Alessandro Raganato** — who kept our community informed with timely updates about all aspects of the conference. The huge number of emails and communications that we received was smoothly managed throughout the year by our Internal Communication Chairs, **Sara Tonelli** and **Yiquan Wu**. We are very grateful!

I am particularly proud of the creation of a new role, the Documentation Chair, taken by **Chenghua Lin**, who did strenuous work to assemble a new handbook with up-to-date information coming from all chairs — whom I also thank for their valuable contributions — and maintain a clear and consistent record of all our processes. Given the speed at which all conference roles evolve in their duties and commitments, I hope this role will become standard in upcoming conferences to ensure a smooth transition.

Last but not least, I wish to thank the ACL Executive Committee for their guidance, and my fellow past General and Program Chairs, and especially **Claire Gardent** and **Colin Cherry**, for generously sharing their experience and insights.

ACL 2025 is a community-wide effort. Whether you are presenting a paper, leading a workshop or tutorial, volunteering, mentoring, supporting or attending to learn and connect, your presence and participation make a difference at this conference. Thank you for being part of it!

Welcome to Vienna. Welcome to ACL 2025!

Roberto Navigli
Sapienza University of Rome & Babelscape

General Chair, ACL 2025

Message from the Program Chairs

Welcome to the 63rd Annual Meeting of the Association for Computational Linguistics!

ACL 2025 will be held in a hybrid format, offering attendees the option to join us in person in vibrant Vienna, Austria, or to participate remotely from anywhere in the world. Organizing ACL 2025 has been a collaborative effort, made possible by the dedication and hard work of thousands of people. We gratefully acknowledge the support and contributions of the following people:

- The General Chair, Roberto Navigli;
- The ARR Editors-in-Chief of the February 2025 cycle (Jun Suzuki, Jing Jiang, and Xiaodan Zhu) and the entire team (Mausam, Viviane Moreira, Vincent Ng, Lilja Øvrelid, Anna Rogers, Michael White, Margot Mieskes, Sarvnaz Karimi);
- The techical OpenReview chairs, Niket Tandon, Lizhen Qu, and the OpenReview support team, in particular Rachel, for multiple rounds of technical help in setting up ACL 2025 on the Open Review platform;
- The 169 Senior Area Chairs;
- The 1,937 Area Chairs and the 11,720 reviewers;
- The best paper committee chairs, Rada Mihalcea and Roi Reichart, and the best paper committee members;
- The ethics chairs, Karën Fort and Bjorn Ross;
- The workshop chairs, Terra Blevins and Christophe Gravier;
- The tutorial chairs, Yuki Arase, David Jurgens and Fei Xia;
- The industry track chairs, Yunyao Li and Georg Rehm;
- The demonstration chairs, Pushkar Mishra, Smaranda Muresan, and Tao Yu;
- The internal communications chairs, Sara Tonelli and Yiquan Wu;
- The website and conference app chairs, Xudong Han and Alessandro Raganato;
- The publication chairs, Pierpaolo Basile, Libo Qin, and Zhenghao Liu;
- The handbook chairs, Els Lefever and Qiongkai Xu;
- The local organization chairs, Benjamin Roth and Dagmar Gromann, and their team;
- The visa chairs, Rexhina Blloshmi and Eleni Ilkou;
- The publicity and social media chairs, Anette Frank, Shruti Rijhwani and Horacio Saggion;
- The documentation chair, Chenghua Lin;
- The student research workshop chairs, Zhu Liu, Mingyang Wang and Jin Zhao;
- The student research workshop chairs faculty advisors, Lea Frermann, Daniel Hershcovich and Tristan Miller;
- The student volunteer chairs, Pedro Henrique Luz de Araujo and Eleonora Mancini;

- The diversity and inclusion chairs, Senja Pollak, Maria Ryskina, Shane Storks and Hwaran Lee;
- The sponsorship chairs, Raffaella Bernardi and Thomas Scialom;
- The virtual infrastructure chairs, Manling Li, Yang Liu and Avi Sil;
- The ACL Anthology Director Matt Post and his team;
- The TACL editors-in-chief (Asli Celikyilmaz, Roi Reichart, Dilek Hakkani-Tur) and CL Editor in-Chief Wei Lu for coordinating TACL and CL presentations with us;
- The ACL 2024 Program Chairs, Lun-Wei Ku, André F. T. Martins, Vivek Srikumar, for information and support;
- Damira Mrsic and Underline Team;
- Jennifer Rachford and entire conference support staff;
- All the authors of papers who submitted their papers for review in the ARR 2025 February cycle and those who committed to the ACL 2025 conference.

Review Process

All submissions to ACL 2025 went through a two-stage review process. First, papers were submitted to the ACL Rolling Review (ARR), where they were reviewed by reviewers and received meta-reviews from Area Chairs. Then, authors had the option to commit their reviewed papers to ACL via a separate ACL 2025 commitment site. At this stage, Senior Area Chairs provided recommendations, and final acceptance decisions were made by the Program Chairs. This process is consistent with previous conferences, including ACL 2024, EACL 2024, and NAACL 2025.
We worked closely with the ARR team, particularly the ARR February 2025 Editors-in-Chief, and served as guest Editors-in-Chief for this round. We helped recruit new reviewers and Area Chairs to ARR, resulting in 11,720 reviewers and 1,942 Area Chairs in the February 2025 ARR cycle to which most ACL 2025 papers were submitted. ACL also recruited 169 Senior Area Chairs to oversee the review and meta-review process. Overall, the ARR process ran smoothly, ensuring that all submitted papers received at least three reviews and a meta-review. For the ACL commitment phase, Senior Area Chairs made recommendations for 5,356 committed papers based on the reviews, meta-reviews, and the papers themselves, with final acceptance decisions made by the Program Chairs.

Acceptance Rate

In total, there are 1,699 papers accepted to the Main Conference and 1,392 papers accepted to Findings. The acceptance rate calculation follows precedent set by previous conferences that go through ACL Rolling Review (ARR), e.g. ACL 2024. The calculation takes into account the multi-stage process of ARR where a paper may get revised in ARR and then later committed to the conference. In total, we had 8,360 unique submissions across the December 2024 and February 2025 ARR cycles of which 5,501 papers were committed to ACL. The acceptance rate is 20.3% for the Main Conference papers and a further 16.7% for Findings papers.

Special Theme: Generalization of NLP Models

ACL 2025's special theme is generalization of NLP models. Generalization is crucial for ensuring that models behave robustly, reliably, and fairly when making predictions on data different from their training data. Achieving good generalization is critically important for models used in real-world applications, as they should emulate human-like behavior. Humans are known for their ability to generalize well, and models should aspire to this standard. The theme track invites empirical and theoretical research

and position and survey papers reflecting on the Generalization of NLP Models. The possible topics of discussion include (but are not limited to) the following:

1. How can we enhance the generalization of NLP models across various dimensions—compositional, structural, cross-task, cross-lingual, cross-domain, and robustness?
2. What factors affect the generalization of NLP models?
3. What are the most effective methods for evaluating the generalization capabilities of NLP models?
4. While Large Language Models (LLMs) significantly enhance the generalization of NLP models, what are the key limitations of LLMs in this regard?

We received 128 submissions to the theme track during the review phase. Among these, 41 papers were accepted to the main conference and a further 33 to Findings of ACL 2025. The conference will also feature a panel discussion on the theme of Generalization, with the participation of leading experts in this area.

Best Paper Selection

ACL 2025 implemented the updated ACL award policy that seeks to expand the pool of work recognized as outstanding. In total 117 papers were nominated by the reviewers, area chairs and senior area chairs for best paper consideration. The best paper committee assessed these papers to select the best papers (featuring $(\leq)0.6\%$ of accepted papers), outstanding papers (featuring $(\leq)2.5\%$ of accepted papers), and special awards for social impact and best resource. Based on the review by the Best Paper committee, 39 papers have been selected for awards in the above categories. Separately, the senior area chairs also nominated their favourite papers as SAC Highlights.
In addition, we have awards for test of time award for a paper published in TACL in 2013 or 2014 and the best paper award for a paper published in TACL in 2024. The final selection was made by the best paper committee, and the winners will be announced during the closing ceremony. The ACL 2025 Best Papers will also be given an opportunity to present their work in the closing ceremony.

Program Composition & Presentation Modes

Based on feedback from the conference support staff and the Underline team after ACL 2025, we decided to hold the virtual presentations sessions during the main conference. This enables us to align the virtual sessions with time slots when in-person participants are available. This approach allows virtual attendees to participate concurrently with the physical event, avoiding the need for organizers and attendees to engage with the conference twice and separately.
This year, 218 main conference papers were selected for oral presentations by the program chairs, with the goal of creating a well-rounded program featuring a diverse set of topics instead of selecting papers based on their review scores. This year we have introduced a panel section of the conference. Out of the selected oral presenters, 25 will have an opportunity to not only present their work but also participate in a panel discussion. We have lined up five panels on the theme of: Generalisation in NLP, LLM alignment, Human-centred NLP, Interpretability and model analysis, Multilinguality and language diversity.
In addition to the main conference papers, the ACL program also includes 18 papers accepted by Computational Linguistics and 42 papers accepted by Transactions of the ACL (TACL). Among these, 11 journal papers will be presented in-person as oral presentations, thematically distributed across appropriate sessions.
This year the conference will also for the first time feature a dedicated Findings poster session with reception. All Findings presenters have been assigned poster presentations in this session or alongside other posters for main conference papers in the same track. Rounding out the program are dedicated sessions for the demonstrations track and the student research workshop.

Keynotes and Panel This year's program features an impressive lineup of two keynote presentations:

- **Prof. Luke Zettlemoyer** from Paul G. Allen School of Computer Science & Engineering at the University of Washington, and a Senior Research Director at Meta, will share his insights on "*Rethinking Pretraining: Data and Architecture*."
- **Prof. Verena Rieser** from Google DeepMind will present on "*Whose Gold? Re-imagining Alignment for Truly Beneficial AI*."

Alongside these keynotes, we are thrilled to host a panel discussion aiming to answer the question *Can large language models (LLMs) generalize?*. Our esteemed panelists include:

- **Prof. Eduard Hovy**, University of Melbourne (who will also act as the panel chair).
- **Prof. Mirella Lapata**, University of Edinburgh
- **Prof. Yue Zhang**, Westlake University
- **Prof. Dan Roth**, University of Pennsylvania and Oracle

This diverse group of panelists will provide a comprehensive view of the latest trends and challenges in Generalisation of NLP in the Large Language Models era. We hope you enjoy this year's diverse and engaging program!

Ekaterina Shutova (University of Amsterdam and Stanford University)
Mohammad Taher Pilehvar (Cardiff University and Tehran Institute for Advanced Studies)
Joyce Nakatumba-Nabende (Makerere University)
Wanxiang Che (Harbin Institute of Technology)

ACL 2025 Program Co-Chairs

Organizing Committee

General Chair

Roberto Navigli, Sapienza University Rome & Babelscape

Program Chairs

Wanxiang Che, Harbin Institute of Technology
Joyce Nabende, Makerere University
Ekaterina Shutova, University of Amsterdam
Mohammad Taher Pilehvar, Cardiff University / Tehran Institute for Advanced Studies

Local Organization

Benjamin Roth, University of Vienna
Dagmar Gromann, University of Vienna

Workshop Chairs

Terra Blevins, University of Vienna
Christophe Gravier, Université Jean Monnet

Tutorial Chairs

Yuki Arase, Tokyo Institute of Technology
David Jurgens, University of Michigan
Fei Xia, University of Washington

Demonstration Chairs

Pushkar Mishra, Google DeepMind / Cambridge
Smaranda Muresan, Barnard College, Columbia University
Tao Yu, The University of Hong Kong

Student Research Workshop Chairs

Zhu Liu, Tsinghua University
Mingyang Wang, LMU
Jin Zhao, Brandeis University

Student Research Workshop Chairs: Faculty Advisors

Lea Frermann, University of Melbourne
Daniel Hershcovich, University of Copenhagen
Tristan Miller, University of Manitoba

Publication Chairs

Pierpaolo Basile, University of Bari
Libo Qin, Central South University
Zhenghao Liu, Northeastern University

Handbook Chair

Els Lefever, Ghent University
Qiongkai Xu, Macquarie University

Sponsorship Chairs

Raffaella Bernardi, University of Trento
Thomas Scialom, Meta AI

Diversity and Inclusion Chairs

Senja Pollak, Jožef Stefan Institute
Maria Ryskina, MIT
Shane Storks, University of Michigan
Hwaran Lee, NAVER

Publicity and Social Media Chairs

Anette Frank, Heidelberg University
Shruti Rijhwani, Google DeepMind
Horacio Saggion, Pompeu Fabra University

Website and Conference App Chairs

Xudong Han, LibrAI / MBZUAI
Alessandro Raganato, University of Milano-Bicocca

Ethics Chairs

Karën Fort, Université de Lorraine / LORIA
Bjorn Ross, University of Edinburgh

Student Volunteer Chairs

Pedro Henrique Luz de Araujo, University of Vienna
Eleonora Mancini, University of Bologna

Technical Open Review Chairs

Niket Tandon, Allen AI
Lizhen Qu, Monash University

Virtual Infrastructure Chair

Manling Li, Northwestern University

Yang Liu, Microsoft
Avil Sil, IBM Research

Internal Communication Chairs

Sara Tonelli, FBK
Yiquan Wu, Zhejiang University

Industry Track Chairs

Yunyao Li, Adobe
George Rehm, DFKI

Best Paper Committee Chairs

Rada Mihalcea, University of Michighan
Roi Reichart, Technion - Israel Institute of Technology

Visa Chairs

Rexhina Blloshmi, Amazon AGI
Eleni Ilkou, University of Hannover

Documentation Chairs

Chenghua Lin, University of Manchester

ARR Editors in Chief

Jun Suzuki, Tohoku University
Jing Jiang, Australian National University
Xiaodan Zhu, Queen's University

Program Committee

Senior Area Chairs

David Ifeoluwa Adelani, McGill University
Alan Akbik, Humboldt Universität Berlin
Raquel Alhama, University of Amsterdam
Afra Alishahi, Tilburg University
Jacob Andreas, Massachusetts Institute of Technology
Maria Antoniak, Allen Institute for Artificial Intelligence
Marianna Apidianaki, University of Pennsylvania
Jun Araki, Bosch Research
Ehsaneddin Asgari, Qatar Computing Research Institute
Edoardo Barba, Sapienza University of Rome
Jasmijn Bastings, Google DeepMind
Beata Beigman Klebanov, Educational Testing Service
Ahmad Beirami, Google DeepMind
Yonatan Belinkov, Technion
Meriem Beloucif, Uppsala University
Jonathan Berant, Tel Aviv University
Steven Bethard, University of Arizona
Alexandra Birch-Mayne, University of Edinburgh
Jelke Bloem, University of Amsterdam
Danushka Bollegala, University of Liverpool
Houda Bouamor, Carnegie Mellon University
Jordan Boyd-Graber, University of Maryland
Emanuele Bugliarello, Google DeepMind
Jan Buys, University of Cape Town
Aoife Cahill, Dataminr
Jose Camacho-Collados, Cardiff University
Hanjie Chen, Rice University
Mingda Chen, Meta FAIR
Danqi Chen, Princeton University
Christos Christodoulopoulos, Amazon
Arman Cohan, Yale University
Simone Conia, Sapienza University of Rome
Ryan Cotterell, Swiss Federal Institute of Technology
Yiming Cui, iFLYTEK Research
Raj Dabre, National Institute of Information and Communications Technology (NICT)
Tejaswini Deoskar, Utrecht University
Leon Derczynski, University of Copenhagen
Yuxiao Dong, Tsinghua University
Li Dong, Microsoft Research
Yue Dong, University of California, Riverside
A. Seza Doğruöz, Ghent University
Greg Durrett, University of Texas at Austin
Ondřej Dušek, Charles University, Prague
Nouha Dziri, Allen Institute for Artificial Intelligence
Desmond Elliott, University of Copenhagen
Guy Emerson, University of Cambridge

Alexander Fabbri, Scale AI
Anna Feldman, Montclair State University
Fuli Feng, University of Science and Technology of China
Xiaocheng Feng, Harbin Institute of Technology
Anjalie Field, Johns Hopkins University
Michael Flor, Educational Testing Service
Antske Fokkens, VU University Amsterdam
Zhe Gan, Apple
Albert Gatt, Utrecht University
Mor Geva, Tel Aviv University
Marjan Ghazvininejad, Facebook
Goran Glavaš, Julius-Maximilians-Universität Würzburg
Yvette Graham, Trinity College Dublin
Jiafeng Guo, Chinese Academy of Sciences
Xianpei Han, Institute of Software, CAS
Peter Hase, Schmidt Sciences
Devamanyu Hazarika, Facebook
Yulan He, King's College London
Mohammad Javad Hosseini, Google
Neil Houlsby, Google
Dieuwke Hupkes, Facebook
Ignacio Iacobacci, Elm Europe
Mohit Iyyer, University of Massachusetts
Zhijing Jin, Max Planck Institute and University of Toronto
Shafiq Joty, Nanyang Technological University and SalesForce.com
Rabeeh Karimi Mahabadi, NVIDIA
Frank Keller, University of Edinburgh
Daniel Khashabi, Johns Hopkins University
Najoung Kim, Boston University
Svetlana Kiritchenko, National Research Council Canada
Ekaterina Kochmar, Mohamed bin Zayed University of Artificial Intelligence
Julia Kreutzer, Cohere for AI
Kalpesh Krishna, Google DeepMind
Ranjay Krishna, University of Washington
Matthieu Labeau, Télécom ParisTech
Anne Lauscher, Universität Hamburg
Carolin Lawrence, NEC Laboratories Europe
Wenqiang Lei, Sichuan University
Martha Lewis, University of Amsterdam
Junyi Jessy Li, University of Texas at Austin
Zhenghua Li, Soochow University
Dongxu Li, SalesForce.com
Piji Li, Nanjing University of Aeronautics and Astronautics, China
Yiqun Liu, Tsinghua University
Fangyu Liu, Google DeepMind
Zhiyuan Liu, Tsinghua University
Kang Liu, Institute of Automation, Chinese Academy of Sciences
Edison Marrese-Taylor, National Institute of Advanced Industrial Science and Technology (AIST)
Joshua Maynez, Google
Dipendra Misra, Mosaic Research, Databricks
Saif Mohammad, National Research Council of Canada

Nafise Sadat Moosavi, University of Sheffield
Aaron Mueller, Northeastern University
Philippe Muller, IRIT, University of Toulouse
Preslav Nakov, Mohamed bin Zayed University of Artificial Intelligence
Vivi Nastase, University of Geneva
Wolfgang Nejdl, Leibniz University Hannover
Xuefei Ning, Tsinghua University
Debora Nozza, Bocconi University
Giorgio Maria Di Nunzio, University of Padua
Nedjma Ousidhoum, Cardiff University
Alexandros Papangelis, Apple
Gabriella Pasi, University of Milano-Bicocca
Rebecca Passonneau, Pennsylvania State University
Viviana Patti, University of Turin
Hao Peng, University of Illinois Urbana-Champaign
Carla Perez-Almendros, Cardiff University
Tiago Pimentel, ETH Zurich
Edoardo Ponti, University of Edinburgh
Simone Paolo Ponzetto, Universität Mannheim
Daniel Preotiuc-Pietro, Bloomberg
Emily Prud'hommeaux, Boston College
Peng Qi, Orby AI
Marek Rei, Imperial College London
Verena Rieser, Google DeepMind
Roland Roller, German Research Center for AI
Alla Rozovskaya, Queens College, City University of New York
Michael Sejr Schlichtkrull, Queen Mary, University of London
Steven Schockaert, Cardiff University
Rico Sennrich, University of Zurich
Ehsan Shareghi, Monash University
Qinlan Shen, Oracle
Kevin Small, Amazon
Mihai Surdeanu, University of Arizona
Anders Søgaard, Copenhagen University
Chenhao Tan, University of Chicago
Hao Tan, Adobe Systems
Harish Tayyar Madabushi, University of Bath
Alberto Testoni, Amsterdam UMC
Trang Tran, USC Institute for Creative Technologies
Marcos Vinicius Treviso, Instituto de Telecomunicações, Lisbon
Yulia Tsvetkov, Carnegie Mellon University
Daniel Varab, German Research Center for AI
Suzan Verberne, Universiteit Leiden
Karin Verspoor, Royal Melbourne Institute of Technology
Ivan Vulić, University of Cambridge
Henning Wachsmuth, Leibniz Universität Hannover
Byron Wallace, Northeastern University
Qifan Wang, Meta AI
Tong Xiao, Northeastern University
Deyi Xiong, Tianjin University
Ruifeng Xu, Harbin Institute of Technology

Yadollah Yaghoobzadeh, University of Tehran
Diyi Yang, Stanford University
Min Yang, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
Helen Yannakoudakis, King's College London
Huaxiu Yao, University of North Carolina at Chapel Hill
Andrew Yates, Johns Hopkins University
Kai Yu, Shanghai Jiao Tong University
Luke Zettlemoyer, University of Washington and Meta
Weinan Zhang, Harbin Institute of Technology
Meishan Zhang, Harbin Institute of Technology, Shenzhen
Yue Zhang, Westlake University
Qi Zhang, Fudan University
Xin Zhao, Renmin University of China
Yi Zhou, Cardiff University
Arkaitz Zubiaga, Queen Mary University of London

Area Chairs

Omri Abend, Gavin Abercrombie, Heike Adel, Rodrigo Agerri, Ameeta Agrawal, Sweta Agrawal, Ibrahim Said Ahmad, Xi Ai, Alham Fikri Aji, Yamen Ajjour, Md Shad Akhtar, Firoj Alam, Mohammad Aliannejadi, Emily Allaway, Milad Alshomary, Hadi Amiri, Antonios Anastasopoulos, Xiang Ao, Catherine Arnett, Shane Arora, Giuseppe Attanasio, Katherine Atwell

Fan Bai, Jiaxin Bai, Xuefeng Bai, JinYeong Bak, Niranjan Balasubramanian, Timothy Baldwin, Miguel Ballesteros, David Bamman, Sivaji Bandyopadhyay, Xiaoyi Bao, Yu Bao, Roy Bar-Haim, Chitta Baral, Fazl Barez, Jeremy Barnes, Valentin Barriere, Joe Barrow, Alberto Barrón-Cedeño, Max Bartolo, Amrit Singh Bedi, Shabnam Behzad, Adrian Benton, Yevgeni Berzak, Federico Bianchi, Yonatan Bitton, Johannes Bjerva, Terra Blevins, Su Lin Blodgett, Nadav Borenstein, Florian Boudin, Zied Bouraoui, Kevin Bowden, Stephanie Brandl, Pavel Braslavski, Chloé Braud, Davide Buscaldi

Sky CH-Wang, Elena Cabrio, Deng Cai, Xiangrui Cai, Zhaowei Cai, Andrew Caines, Ruken Cakici, Nitay Calderon, Jie Cao, Meng Cao, Meng Cao, Pengfei Cao, Qingqing Cao, Yang Trista Cao, Yixin Cao, Yong Cao, Ziqiang Cao, Doina Caragea, Dallas Card, Dumitru-Clementin Cercel, Soumen Chakrabarti, Saikat Chakraborty, Bharathi Raja Chakravarthi, Chunkit Chan, Angel X Chang, Ernie Chang, Haw-Shiuan Chang, Yi Chang, Snigdha Chaturvedi, Aditi Chaudhary, Chung-Chi Chen, Francine Chen, Guanhua Chen, Hui Chen, Jiangjie Chen, Jifan Chen, Junfan Chen, Kehai Chen, Lu Chen, Muhao Chen, Pin-Yu Chen, Qian Chen, Qianglong Chen, Wei Chen, Wenliang Chen, Xilun Chen, Xinchi Chen, Xinyun Chen, Xiuying Chen, Yangyi Chen, Yidong Chen, Yiming Chen, Yulong Chen, Yun-Nung Chen, Yunmo Chen, Zhiyu Chen, Zhiyu Chen, Zhuang Chen, Fei Cheng, Gong Cheng, Lu Cheng, Pengyu Cheng, Wei Cheng, Xueqi Cheng, Yu Cheng, Jianfeng Chi, Nadezhda Chirkova, Hyunsoo Cho, Sangwoo Cho, Eleanor Chodroff, Edward Choi, Eunsol Choi, Jinho D. Choi, YunSeok Choi, Jaegul Choo, Prafulla Kumar Choubey, Alexandra Chronopoulou, Chenhui Chu, Yi-Ling Chung, Simon Clematide, Shay B Cohen, John M Conroy, Caio Corro, Benoit Crabbé, Heriberto Cuayahuitl, Ruixiang Cui, Aron Culotta

Giovanni Da San Martino, Damai Dai, Bhavana Dalvi Mishra, Rossana Damiano, Sandipan Dandapat, Pradeep Dasigi, Orphee De Clercq, Budhaditya Deb, Pieter Delobelle, Vera Demberg, Daryna Dementieva, Chenlong Deng, Shumin Deng, Yang Deng, Yuntian Deng, Daniel Deutsch, Chris Develder, Kuntal Dey, Jwala Dhamala, Shizhe Diao, Haibo Ding, Kaize Ding, Liang Ding, Liviu P Dinu, Lucia Donatelli, Hanze Dong, Ruihai Dong, Zhicheng Dou, Markus Dreyer, Li Du,

Mengnan Du, Xinya Du, Brian DuSell, Xiangyu Duan, Marie-Catherine de Marneffe

Lukas Edman, Asif Ekbal, Aparna Elangovan, Maha Elbayad, Ahmed Elgohary, Ali Emami, Cristina España-Bonet, Ralph Ewerth

Agnieszka Falenska, Kai Fan, Yaxin Fan, Meng Fang, Marcello Federico, Hao Fei, Naomi Feldman, Jiazhan Feng, Shangbin Feng, Shaoxiong Feng, Shi Feng, Xiachong Feng, Yi Feng, Javier Ferrando, Anjalie Field, Simone Filice, Mark Fishel, Jeffrey Flanigan, Lucie Flek, Radu Florian, Mary Ellen Foster, Andre Freitas, Lea Frermann, Annemarie Friedrich, Yi R. Fung

Saadia Gabriel, Marco Gaido, Matthias Gallé, Niloy Ganguly, Chongming Gao, Yifan Gao, Ze-Feng Gao, Jon Gauthier, Jiahui Geng, Luke Gessler, Abbas Ghaddar, Debanjan Ghosh, Saptarshi Ghosh, Salvatore Giorgi, Mario Giulianelli, Nathan Godey, Seraphina Goldfarb-Tarrant, Omer Goldman, Dan Goldwasser, Sharon Goldwater, Sian Gooding, Sruthi Gorantla, Kyle Gorman, Matthew R. Gormley, Kartik Goyal, Loïc Grobol, Jiuxiang Gu, Qing Gu, Lin Gui, Junliang Guo, Yue Guo, Raghav Gupta, Vivek Gupta, Izzeddin Gur, Bernal Jiménez Gutiérrez, Carlos Gómez-Rodríguez

Nizar Habash, Ivan Habernal, Barry Haddow, Michael Hahn, Dilek Hakkani-tür, Rujun Han, Jie Hao, Mohammed Hasanuzzaman, Shirley Anugrah Hayati, Ben He, Bin He, Bobby He, He He, Jie He, Junxian He, Keqing He, Shizhu He, Tianxing He, Wei He, Daniel Hershcovich, Tosho Hirasawa, Julia Hockenmaier, Valentin Hofmann, Christopher M Homan, Lingzi Hong, Enamul Hoque, Lei Hou, Yufang Hou, Alexander Miserlis Hoyle, Dou Hu, Joy Hu, Joy Hu, Junjie Hu, Linmei Hu, Mengting Hu, Wei Hu, Wenpeng Hu, Xuming Hu, Chen Huang, Jiaxin Huang, Kuan-Hao Huang, Kung-Hsiang Huang, Ruihong Huang, Tenghao Huang, Xinting Huang, Xuanjing Huang, Yi Huang, Yichong Huang, Yinya Huang, Zhenya Huang, Pere-Lluís Huguet Cabot, Chia-Chien Hung, Wonseok Hwang, seung-won hwang

Timour Igamberdiev, Filip Ilievski, Sathish Reddy Indurthi, Naoya Inoue, Radu Tudor Ionescu, Daphne Ippolito, Etsuko Ishii, Hayate Iso, Masaru Isonuma

Matthew Jagielski, Kokil Jaidka, Masoud Jalili Sabet, Shoaib Jameel, Hyeju Jang, Adam Jatowt, Ganesh Jawahar, Akshita Jha, Harsh Jhamtani, Shaoxiong Ji, Tao Ji, Yangfeng Ji, Ziwei Ji, Robin Jia, Zixia Jia, Chao Jiang, Huiqiang Jiang, Zhiying Jiang, Zhuoxuan Jiang, Zhanming Jie, Lifeng Jin, Qiao Jin, Qin Jin, Yohan Jo, Richard Johansson, Wei Ju, Jaap Jumelet, David Jurgens, Preethi Jyothi

Jad Kabbara, Hour Kaing, Anup K. Kalia, Ehsan Kamalloo, Hidetaka Kamigaito, Masahiro Kaneko, Dongyeop Kang, Diptesh Kanojia, Alina Karakanta, Börje F. Karlsson, Pei Ke, Yova Kementchedjhieva, Ilker Kesen, Salam Khalifa, Khalid Al Khatib, Omar Khattab, Hyunjae Kim, Hyunwoo Kim, Jin-Dong Kim, Joo-Kyung Kim, Taeuk Kim, Yeachan Kim, Yunsu Kim, Christo Kirov, Hirokazu Kiyomaru, Dietrich Klakow, Roman Klinger, Christine De Kock, Christine De Kock, Philipp Koehn, Konstantinos Kogkalidis, Keshav Kolluru, Mamoru Komachi, Rik Koncel-Kedziorski, Lingkai Kong, Ioannis Konstas, Michalis Korakakis, Parisa Kordjamshidi, Raphael Koster, Fajri Koto, Emiel Krahmer, Amrith Krishna, Kundan Krishna, Shivani Kumar, Anoop Kunchukuttan, Souvik Kundu, Tatsuki Kuribayashi, Shuhei Kurita

Wai Lam, Wuwei Lan, Yunshi Lan, Md Tahmid Rahman Laskar, Hung Le, Thai Le, Eric Le Ferrand, Andrew Lee, Chia-Hsuan Lee, Dong-Ho Lee, Hung-yi Lee, Hwanhee Lee, Hwaran Lee, Jaeho Lee, Jay-Yoon Lee, Roy Ka-Wei Lee, SangKeun Lee, Seolhwa Lee, Oliver Lemon, Heather Lent, Michael A. Lepori, Kristina Lerman, Piyawat Lertvittayakumjorn, Chenliang Li, Dongfang

Li, Fei Li, Jing Li, Jingjing Li, Jiyi Li, Junhui Li, Junyi Li, Lei Li, Lei Li, Linyang Li, Manling Li, Mingxiao Li, Peifeng Li, Peng Li, Qian Li, Ru Li, Ruizhe Li, Shuai Li, Tao Li, Wei Li, Wenjie Li, Xiang Lorraine Li, Xiang Li, Xiaoli Li, Xin Li, Xintong Li, Yaliang Li, Yanzeng Li, Yaoyiran Li, Yinghui Li, Yingjie Li, Yu Li, Yuan-Fang Li, Zhuang Li, Zixuan Li, Zixuan Li, Bin Liang, Paul Pu Liang, Jindřich Libovický, Constantine Lignos, Chenghua Lin, Hongfei Lin, Hsien-chin Lin, Nankai Lin, Shou-De Lin, Yankai Lin, Yong Lin, Zhaojiang Lin, Pierre Lison, Bang Liu, Bingquan Liu, Dongqi Liu, Fenglin Liu, Guangliang Liu, Hao Liu, Hao Liu, Hui Liu, Jiahao Liu, Ming Liu, Qi Liu, Qianying Liu, Qun Liu, Tianyu Liu, Tingwen Liu, Wei Liu, Xiao Liu, Xiao Liu, Xuebo Liu, Yang Liu, Ye Liu, Yixin Liu, Zhenghao Liu, Zhiyuan Liu, Zuozhu Liu, Karen Livescu, Shayne Longpre, Adam Lopez, Hongyuan Lu, Wei Lu, Weiming Lu, Yaojie Lu, Yaojie Lu, Li Lucy, Thomas Lukasiewicz, Michal Lukasik, Junyu Luo, Xiao Luo, Anh Tuan Luu, Ang Lv

Fenglong Ma, Jing Ma, Mingyu Derek Ma, Ruotian Ma, Xiao Ma, Yubo Ma, Manuel Mager, Kyle Mahowald, Valentin Malykh, Potsawee Manakul, Xin Mao, Zhendong Mao, David Mareček, Katerina Margatina, Sérgio Matos, Jonathan May, Sahisnu Mazumder, Ninareh Mehrabi, Clara Meister, Dheeraj Mekala, Zaiqiao Meng, Sabrina J Mielke, Simon Mille, Kanishka Misra, Fengran Mo, Daichi Mochihashi, Ashutosh Modi, Hosein Mohebbi, Tasnim Mohiuddin, Diego Molla, Masoud Monajatipoor, Syrielle Montariol, Manuel Montes, Roser Morante, Makoto Morishita, David R. Mortensen, Marius Mosbach, Lili Mou, Youssef Mroueh, Rudra Murthy

Masaaki Nagata, Linyong Nan, Jason Naradowsky, Pranav Narayanan Venkit, Usman Naseem, Mir Tafseer Nayeem, Isar Nejadgholi, Hwee Tou Ng, Dat Quoc Nguyen, Dong Nguyen, Hoang H Nguyen, Minh-Tien Nguyen, Ansong Ni, Shiwen Ni, Garrett Nicolai, Jan Niehues, Joel Niklaus, Mitja Nikolaus, Qiang Ning, Takashi Ninomiya, Tong Niu, Farhad Nooralahzadeh, Pierre Nugues

Maciej Ogrodniczuk, Alice Oh, Naoaki Okazaki, Manabu Okumura, Alexandra Olteanu, Arturo Oncevay, Matan Orbach, Yohei Oseki, Simon Ostermann, Jessica Ouyang, Mete Ozay

Maria Leonor Pacheco, Aishwarya Padmakumar, Sebastian Padó, Serguei V. S. Pakhomov, Liangming Pan, Ashwinee Panda, Liang Pang, Isabel Papadimitriou, Sara Papi, Letitia Parcalabescu, Cecile Paris, Chanjun Park, Joonsuk Park, Kunwoo Park, Roma Patel, Siddharth Patwardhan, Debjit Paul, Ellie Pavlick, Hao Peng, Hao Peng, Minlong Peng, Siyao Peng, Veronica Perez-Rosas, Sandro Pezzelle, Chau Thi Minh Pham, Chau Thi Minh Minh Pham, Yuval Pinter, Irina Piontkovskaya, Benjamin Piwowarski, Bryan A. Plummer, Soujanya Poria, Shrimai Prabhumoye, Animesh Prasad, Radityo Eko Prasojo, Emily Prud'hommeaux, Danish Pruthi

Tao Qi, Chen Qian, Cheng Qian, Tieyun Qian, Libo Qin, Yanxia Qin, Chen Qiu, Han Qiu, Haoyi Qiu, Jielin Qiu, Xipeng Qiu, Lizhen Qu, Tingyu Qu, Xiaojun Quan

Dinesh Raghu, Vyas Raina, Hossein Rajaby Faghihi, Dhananjay Ram, Deepak Ramachandran, Carlos Ramisch, Alan Ramponi, Leonardo Ranaldi, Tharindu Ranasinghe, Surangika Ranathunga, Yanghui Rao, Hannah Rashkin, Charvi Rastogi, Shauli Ravfogel, Jishnu Ray Chowdhury, Anton Razzhigaev, Traian Rebedea, Ehud Reiter, Navid Rekabsaz, Haopeng Ren, Ruiyang Ren, Yi Ren, Zhaochun Ren, Leonardo F. R. Ribeiro, Anthony Rios, Julia Romberg, Salvatore Romeo, Michael Roth, Joseph Le Roux, Frank Rudzicz, Ramon Ruiz-Dolz, Paul Röttger

Mrinmaya Sachan, Mehrnoosh Sadrzadeh, Kenji Sagae, Horacio Saggion, Gaurav Sahu, Hassan Sajjad, Elizabeth Salesky, Shailaja Keyur Sampat, Maarten Sap, Abulhair Saparov, Gabriele Sarti, Asad B. Sayeed, Yves Scherrer, Viktor Schlegel, Hinrich Schuetze, Hendrik Schuff, Raphael Schumann, João Sedoc, Indira Sen, Shubhashis Sengupta, Maureen De Seyssel, Lei Sha, Guokan Shang, Jingbo Shang, Ori Shapira, Pratyusha Sharma, Vasu Sharma, Hassan Shavarani, Ravi

Shekhar, Jiaming Shen, Xiaoyu Shen, Yongliang Shen, Peng Shi, Tianze Shi, Zhengliang Shi, Jiasheng Si, Qingyi Si, Carina Silberer, Edwin Simpson, Maneesh Kumar Singh, Koustuv Sinha, Sunayana Sitaram, Konstantinos Skianis, Noah A. Smith, Wei Song, Xingyi Song, Yangqiu Song, Yixiao Song, Sandeep Soni, Aitor Soroa, Gerasimos Spanakis, Balaji Vasan Srinivasan, Shashank Srivastava, Jacopo Staiano, Gabriel Stanovsky, Elias Stengel-Eskin, Kevin Stowe, Marco Antonio Stranisci, Jinsong Su, Qi Su, Qinliang Su, Yu Su, Yusheng Su, Saku Sugawara, Alessandro Suglia, Dianbo Sui, Elior Sulem, Aixin Sun, Fei Sun, Hao Sun, Jiashuo Sun, Kai Sun, Lichao Sun, Mingming Sun, Ruoxi Sun, Yuan Sun, Ziteng Sun, Shahbaz Syed, Michal Štefánik

Santosh T.Y.S.S, Hiroya Takamura, Zeerak Talat, Estefania Talavera, Minghuan Tan, Zhaoxuan Tan, Zhixing Tan, Gongbo Tang, Hao Tang, Hongyin Tang, Siliang Tang, Tianyi Tang, Xiangru Tang, Zhiyang Teng, Chris Thomas, Jesse Thomason, Zhiliang Tian, Maria Tikhonova, Ivan Titov, Amalia Todirascu, Antonio Toral, Kristina Toutanova, Amine Trabelsi, Quan Hung Tran, Chen-Tse Tsai, Richard Tzong-Han Tsai, Reut Tsarfaty, Yu-Hsiang Tseng, Yoshimasa Tsuruoka, Jingxuan Tu, Kewei Tu, Yunbin Tu, Zhaopeng Tu, Yi-Lin Tuan, Gokhan Tur, Ferhan Ture, Martin Tutek

Ana Sabina Uban, Adaku Uchendu

Esaú VILLATORO-TELLO, Neeraj Varshney, Alakananda Vempala, Pat Verga, Gaurav Verma, Serena Villata, Juraj Vladika, Rob Voigt, Thang Vu, Thuy Vu, Tu Vu, Yogarshi Vyas

V. W., Xiaojun Wan, Xingchen Wan, Benyou Wang, Bin Wang, Boshi Wang, Boxin Wang, Chengyu Wang, Fei Wang, Guoyin Wang, Hao Wang, Haoyu Wang, Jiaan Wang, Jin Wang, Jingjing Wang, Keze Wang, Liang Wang, Lingzhi Wang, Liwei Wang, Lucy Lu Wang, Peifeng Wang, Renzhi Wang, Rui Wang, Shijin Wang, Shuohang Wang, Siyuan Wang, Weiqi Wang, Wenjie Wang, Wenjie Wang, Wenya Wang, Xiaolei Wang, Xiaozhi Wang, Xin Eric Wang, Xin Wang, Xindi Wang, Xinyu Wang, Xuan Wang, Yan Wang, Yile Wang, Yu Wang, Zhaowei Wang, Zhongqing Wang, Zifeng Wang, Zihao Wang, Alex Warstadt, Shinji Watanabe, Taro Watanabe, Kellie Webster, Wei Wei, Shira Wein, Nathaniel Weir, Leonie Weissweiler, Haoyang Wen, Derry Tanti Wijaya, Thilini Wijesiriwardene, Genta Indra Winata, Dustin Wright, Bingzhe Wu, Chien-Sheng Wu, Chuhan Wu, Fangzhao Wu, Hongqiu Wu, Lijun Wu, Qiyu Wu, Xian Wu, Xiao-Ming Wu, Xiaobao Wu, Yiquan Wu, Yuanbin Wu, Yunfang Wu, Zhen Wu, Zhuofeng Wu

Heming Xia, Qianqian Xie, Tengyang Xie, Chenyan Xiong, Bo Xu, Chunpu Xu, Han Xu, Hongfei Xu, Jiacheng Xu, Jinan Xu, Qiongkai Xu, Silei Xu, Tong Xu, Xinnuo Xu, Yumo Xu, Zhaozhuo Xu, Nianwen Xue

Shuntaro Yada, Shweta Yadav, Ikuya Yamada, Ikuya Yamada, An Yan, Hanqi Yan, Jianhao Yan, Jun Yan, Lingyong Yan, Rui Yan, Hitomi Yanaka, Chao-Han Huck Yang, Deqing Yang, Hao Yang, Jian Yang, Jie Yang, Jingkang Yang, Mingming Yang, Qian Yang, Shaohua Yang, Songlin Yang, Xianjun Yang, Yezhou Yang, Zhenglu Yang, Zhiwei Yang, Zonglin Yang, Bingsheng Yao, Ziyu Yao, Zonghai Yao, Gregory Yauney, Fanghua Ye, Hai Ye, Jinyoung Yeo, Seid Muhie Yimam, Da Yin, Jian Yin, Yongjing Yin, Kang Min Yoo, Naoki Yoshinaga, Jiaxuan You, Jianfei Yu, Jianxing Yu, Juntao Yu, Lei Yu, Liang-Chih Yu, Tong Yu, Wenhao Yu, Youngjae Yu, Yue Yu, Zheng Yuan, Zheng Yuan, Xiang Yue, Zhenrui Yue

Fengbin ZHU, Fabio Massimo Zanzotto, Xingshan Zeng, Torsten Zesch, Chen Zhang, Chen Zhang, Hao Zhang, Haopeng Zhang, Hongyang Zhang, Jiajun Zhang, Jiaxin Zhang, Jing Zhang, Kai Zhang, Lei Zhang, Lu Zhang, Ningyu Zhang, Rongzhi Zhang, Rui Zhang, Ruiyi Zhang, Shaolei Zhang, Shichang Zhang, Shikun Zhang, Shiyue Zhang, Shuai Zhang, Shuai Zhang, Taolin

Zhang, Tong Zhang, Wei Zhang, Weinan Zhang, Wenqi Zhang, Wenxuan Zhang, Xinsong Zhang, Yan Zhang, Yang Zhang, Yang Zhang, Yi Zhang, Yi Zhang, Yiming Zhang, Ying Zhang, Yong Zhang, Yu Zhang, Yu Zhang, Yuhui Zhang, Zeyu Zhang, Zhe Zhang, Zheng Zhang, Zhengyan Zhang, Zhenyu Zhang, Zhuosheng Zhang, Chao Zhao, Chen Zhao, Chenye Zhao, Dongyan Zhao, Jieyu Zhao, Jinming Zhao, Jun Zhao, Runcong Zhao, Tiejun Zhao, Xinran Zhao, Xuandong Zhao, Yang Zhao, Yanpeng Zhao, Zhixue Zhao, Chujie Zheng, Zaixiang Zheng, Zilong Zheng, Ben Zhou, Guangyou Zhou, Hanzhang Zhou, Jiawei Zhou, Jie Zhou, Kaitlyn Zhou, Qingyu Zhou, Tianyi Zhou, Wenxuan Zhou, Xiang Zhou, Yichao Zhou, Yucheng Zhou, Hao Zhu, Wanrong Zhu, Wenhao Zhu, Yilun Zhu, Zining Zhu, Terry Yue Zhuo, Shi Zong, Yicheng Zou, hai zhao

Reviewers

Aakanksha, Asma Ben Abacha, Micheal Abaho, Mahdi Abavisani, Ahmed Abbasi, Mohamed Abdalla, Abdelrahman Abdallah, Mohamed Abdellatif, Muhammad Abdul-mageed, Amir Abdullah, Badr M Abdullah, Idris Abdulmumin, Giuseppe Abrami, Mitchell Abrams, Abdalghani Abujabal, Artem Abzaliev, Lasha Abzianidze, Arkadeep Acharya, Sharada Acharya, Emre Can Acikgoz, Gowthamkrishna Addluri, Tosin Adewumi, Suman Adhya, Somak Aditya, Vaibhav Adlakha, Muhammad Abdullah Adnan, Helena Gomez Adorno, Estelle Aflalo, Haithem Afli, Aishwarya Agarwal, Amit Agarwal, Ankush Agarwal, Saurabh Agarwal, Siddhant Agarwal, Ameya Agaskar, Arshiya Aggarwal, Divyanshu Aggarwal, Karan Aggarwal, Milan Aggarwal, Mohamed Aghzal, Eneko Agirre, Ayush Agrawal, Samarth Agrawal, Carlos Alejandro Aguirre, N.nasurudeen Ahamed, Sina Ahmadi, Arash Ahmadian, Murtadha Ahmed, Shafiuddin Rehan Ahmed, Zeeshan Ahmed, Jaewoo Ahn, Sumyeong Ahn, Kian Ahrabian, Sanchit Ahuja, Lin Ai, Qingyao Ai, Wei Ai, Ankit Aich, Taichi Aida, Laura Aina, Akiko Aizawa, Pritom Saha Akash, Mohammad Akbartajari, Mohammad Akbari, Aditya Akella, Benjamin Akera, Mahmud Elahi Akhter, Berfin Aktas, Syeda Sabrina Akter, Hend Al-khalifa, Hadeel Al-negheimish, Nora Al-twairesh, Jesujoba Oluwadara Alabi, Ozge Alacam, Mehwish Alam, Lisa Alazraki, Hanan Aldarmaki, Anton Mikhailovich Alekseev, Laura Alonso Alemany, Georgios Alexandridis, David Alfter, Israa Alghanmi, Bashar Alhafni, Alnur Ali, Felermino D. M. A. Ali, Iqra Ali, Sarwan Ali, Ilseyar Alimova, Pegah Alipoormolabashi, Dan Alistarh, Badr Alkhamissi, Hani Alomari, Iñigo Alonso, Kenneth Alperin, Sawsan Alqahtani, Hilal Alquabeh, Ana Alves, Duarte Miguel Alves, Fakhraddin Alwajih, Zaid Alyafeai, Mahsa Amani, Alfonso Amayuelas, Maxime Amblard, Enrique Amigo, Muhammad Saad Amin, Saadullah Amin, Silvio Amir, Samuel Joseph Amouyal, Bang An, Chenxin An, Haozhe An, Kaikai An, Na Min An, Shengnan An, Wenbin An, Zhenwei An, Aliki Anagnostopoulou, Ashish Anand, Amith Ananthram, Judith Jeyafreeda Andrew, Nicholas Andrews, Ion Androutsopoulos, Piotr Andruszkiewicz, Rico Angell, Krasimir Angelov, Tatiana Anikina, Sifatul Anindho, Murali Annavaram, Miriam Anschütz, Wang Ante, Jacy Reese Anthis, Diego Matteo Antognini, Dimosthenis Antypas, David Anugraha, Abrar Anwar, Saba Anwar, Junyi Ao, Tatsuya Aoyama, Srikar Appalaraju, Mario Ezra Aragón, Eiji Aramaki, Oscar Araque, Pedro Henrique Luz De Araujo, Jerónimo Arenas-garcía, Arturo Argueta, Esteban Garces Arias, Samee Arif, Nurul Fajrin Ariyani, Konstantine Arkoudas, Jordi Armengol-estapé, Akshatha Arodi, Akhil Arora, Arnav Arora, Gaurav Arora, Sanchit Arora, Siddhant Arora, Ines Arous, David Arps, Mousa Arraf, Katya Artemova, Mikel Artetxe, Keren Artiaga, Mohammad Arvan, Bibek Aryal, Daiki Asami, Yuki Asano, Nicholas Asher, Saleh Ashkboos, Joshua Ashkinaze, Salena Torres Ashton, Dennis Assenmacher, Md. Atabuzzaman, Duygu Ataman, Thushari Atapattu, Yash Kumar Atri, Omar Attia, Joseph Attieh, Berk Atıl, P. V. S. Avinesh, Abinew Ali Ayele, Ansar Aynetdinov, Hammad Ayyubi, Amos Azaria, Israel Abebe Azime, Abdul Aziz, Dilshod Azizov, Gorka Azkune, Daichi Azuma, Matthias Aßenmacher

Premjith B, Yukino Baba, Nikolay Babakov, Susanna Maria Baby, Stephen Bach, Sher Badshah,

Ingeol Baek, Jinheon Baek, Roxanne El Baff, Niyati Bafna, Mahdieh Baghshah, Seyed Ali Bahrainian, Ge Bai, Haoli Bai, He Bai, Jiaqi Bai, Jun Bai, Long Bai, Xiang Bai, Xiaopeng Bai, Yimeng Bai, Yu Bai, Yu Bai, Yuelin Bai, Yuyang Bai, Yuzhuo Bai, Divya Jyoti Bajpai, Pradeep Bajracharya, Gurkirat Singh Bajwa, George Arthur Baker, Yavuz Faruk Bakman, Oana Denisa Balalau, Nihal Balani, Ananth Balashankar, Gargi Balasubramaniam, Nishant Balepur, Pier Felice Balestrucci, Wolf-tilo Balke, Dibyanayan Bandyopadhyay, Rabiraj Bandyopadhyay, Sambaran Bandyopadhyay, Somnath Banerjee, Yejin Bang, Naman Bansal, Srijan Bansal, Guangsheng Bao, Jianzhu Bao, Keqin Bao, Rong Bao, Runxue Bao, Siqi Bao, Yuwei Bao, Kfir Bar, Libby Barak, Serguei Barannikov, Alexander Baranov, David La Barbera, M Saiful Bari, Guntis Barzdins, Arslan Basharat, Pierpaolo Basile, Elias Bassani, Elisa Bassignana, Abhipsa Basu, Kinjal Basu, Samyadeep Basu, Riza Batista-navarro, Kayhan Batmanghelich, Anil Batra, Soumya Batra, John Bauer, Tim Baumgärtner, Rachel Bawden, Deniz Bayazit, Adrián Bazaga, Omid Bazgir, Adrien Bazoge, Klaudia Bałazy, Nathanaël Beau, Helard Becerra, Tilman Beck, Karin Becker, Srikanta J. Bedathur, Peter Anthony Beerel, Parishad Behnamghader, Andrei Stefan Bejgu, Gemma Belenguix, Tadesse Destaw Belay, Peter Belcak, Peter Bell, Jonas Belouadi, Tony Belpaeme, Luisa Bentivogli, Uri Berger, Benjamin Bergner, Jordi Bernad, Guillaume Bernard, Timothée Bernard, Raffaella Bernardi, Gabriel Bernier-colborne, Dario Bertero, Amanda Bertsch, Katrien Beuls, Michele Bevilacqua, Marie Bexte, Sourav Sanjukta Bhabesh, Shilpa Bhagavath, Milan Bhan, Manasa Bharadwaj, Rohit K Bharadwaj, Suma Bhat, Hanoz Bhathena, Gagan Bhatia, Rohan Bhatia, Sumit Bhatia, Shaily Bhatt, Amrita Bhattacharjee, Arnab Bhattacharya, Indrajit Bhattacharya, Sourangshu Bhattacharya, Pushpak Bhattacharyya, Bhavya Bhavya, Nikhil Bhendawade, Rajarshi Bhowmik, Amran Bhuiyan, Baolong Bi, Bin Bi, Guanqun Bi, Keping Bi, Ziqian Bi, Laura Biester, Shankar Biradar, Steven Bird, Tegawendé F. Bissyandé, Johanna Björklund, Arno Blaas, Verena Blaschke, Rexhina Blloshmi, Frederic Blum, Boboboliu, Joanne Boisson, Marco Bombieri, Vamshi Krishna Bonagiri, Francis Bond, Claire Bonial, Philipp Borchert, Łukasz Borchmann, Claudia Borg, Hsuvas Borkakoty, Emanuela Boros, Daniel Borrajo, Matteo Bortoletto, Antal Van Den Bosch, Cristina Bosco, Digbalay Bose, Sombit Bose, Abdesselam Bouchekif, Tom Bourgeade, Andrey Bout, Johan Boye, Leonid Boytsov, Murad Bozik, Marco Braga, Maharaj Brahma, David Adam Braude, Daniel Braun, Jannik Brinkmann, Tomás Vergara Browne, Dominique Brunato, Elia Bruni, Mikael Brunila, Alessio Brutti, Christopher Bryant, Xingyuan Bu, Yuqi Bu, Ana-maria Bucur, Kyle Robert Buettner, Nam Khac-hoai Bui, Trung Bui, Andreas Bulling, Laurie Burchell, Sergio Gastón Burdisso, Sophie Burkhardt, Andrea Burns, Mikhail Burtsev, Hendrik Buschmeier, Yash Parag Butala, Miriam Butt, Happy Buzaaba, Bill Byrne, Hoyoon Byun, Sungjoo Byun, Necva Bölücü

Isabel Cachola, Luca Cagliero, Erica Cai, Fengyu Cai, Guohao Cai, Hengyi Cai, Jinglun Cai, Kunlin Cai, Li Cai, Ming Cai, Pengshan Cai, Ruichu Cai, Shu-bin Cai, Weidong Cai, Yi Cai, Yiqing Cai, Yujun Cai, Yuxiang Cai, Zefan Cai, Cristian Buc Calderon, Aylin Caliskan, Iacer Calixto, Jesus Calvillo, Nguyen Cam-tu, Erik Cambria, Necati Cihan Camgoz, Stefano Campese, Duy-cat Can, Marc Canby, Nicola Cancedda, Marie Candito, Biwei Cao, Bochuan Cao, Bowen Cao, Boxi Cao, Hailong Cao, Hannan Cao, Hongye Cao, Jiuxin Cao, Juan Cao, Lang Cao, Liuwen Cao, Minxuan Cao, Qian Cao, Qingxing Cao, Rui Cao, Ruisheng Cao, Shitong Cao, Shuyang Cao, Yanan Cao, Yixuan Cao, Yu Cao, Yu Cao, Yuan Cao, Yunbo Cao, Yupeng Cao, Yuxuan Cao, Zhao Cao, Zhe Cao, Zhiyu Cao, Zouying Cao, Victor Carbune, Ronald Cardenas, Rémi Cardon, Henrique Lopes Cardoso, Luigi Di Caro, Lucien Carroll, Danilo Carvalho, Tommaso Caselli, Silvia Casola, Roman Castagné, Giuseppe Castellucci, Camilla Casula, Paulo Rodrigo Cavalin, Francesco Cazzaro, Jan Cegin, Yukuo Cen, Dong-kyu Chae, Hyungjoo Chae, Joyce Chai, Linzheng Chai, Wenhao Chai, Yekun Chai, Ziwei Chai, Megha Chakraborty, Souradip Chakraborty, Gaël De Chalendar, Alvin Chan, Chimin Chan, David Chan, Hou Pong Chan, Sophia Chan, Christina A Chance, Navin Chandak, Mohit Chandra, Aofei Chang, Buru Chang, Heng Chang, Ma Chang, Maria Chang, Ruidi Chang, Serina Chang, Ting-yun Chang, Tyler A Chang, Yapei Chang,

Yin-wen Chang, Zhiyuan Chang, Wenhan Chao, Yang Chao, Thierry Charnois, Lucas Georges Gabriel Charpentier, Anwoy Chatterjee, Usashi Chatterjee, Akshay Chaturvedi, Sagar Chaturvedi, Duen Chau, Anuj Chauhan, Arnav Santosh Chavan, Kushal Chawla, Muthusamy Chelliah, Ivan Chelombiev, Emmanuel Chemla, Andong Chen, Anthony Chen, Beiduo Chen, Ben Chen, Berlin Chen, Bin Chen, Bingsen Chen, Bo Chen, Bo Chen, Boxing Chen, C. L. Philip Chen, Canyu Chen, Cen Chen, Chacha Chen, Changyu Chen, Chen Chen, Chenhua Chen, Chi Chen, Chien Hung Chen, Congliang Chen, Danlu Chen, Delong Chen, Dingfan Chen, Dongping Chen, Dubing Chen, Enhong Chen, Feng Chen, Gang Chen, Guanjie Chen, Guanyi Chen, Guanzheng Chen, Guiming Hardy Chen, Guizhen Chen, Haibin Chen, Haihua Chen, Hang Chen, Hannah Chen, Hanzhu Chen, Hao Chen, Hao Chen, Hao Chen, Haonan Chen, Hong Chen, Hongyang Chen, Hongyu Chen, Hongyu Chen, Hongzhan Chen, Huiyao Chen, Hung-ting Chen, Jian Chen, Jianda Chen, Jiaqi Chen, Jiawei Chen, Jingchang Chen, Jinghui Chen, Jingyi Chen, Jiuhai Chen, Jixuan Chen, Jiyu Chen, John Chen, Junhao Chen, Junjie Chen, Junying Chen, Junzhe Chen, Kai Chen, Kai Chen, Kai Chen, Kaiyuan Chen, Kang Chen, Kangjie Chen, Kedi Chen, Kejia Chen, Kejiang Chen, Kuang-ming Chen, Lei Chen, Leshang Chen, Liang Chen, Lihu Chen, Lin Chen, Lizhi Chen, Lizhong Chen, Long Chen, Longze Chen, Luoxin Chen, Meiqi Chen, Meng Chen, Mouxiang Chen, Nuo Chen, Nuo Chen, Nuo Chen, Patrick Chen, Pinzhen Chen, Po-chun Chen, Qian Chen, Qiguang Chen, Qin Chen, Qingcai Chen, Qizhou Chen, Qun Chen, Ruijun Chen, Ruirui Chen, Ruizhe Chen, Shan Chen, Shijie Chen, Shisong Chen, Shiwei Chen, Shuo Chen, Simin Chen, Sishuo Chen, Tianlong Chen, Tianxiang Chen, Tong Chen, Tong Chen, Tongfei Chen, Valerie Chen, Wei Chen, Wei Chen, Wei-fan Chen, Weidong Chen, Weize Chen, Wenqing Chen, Wenting Chen, Wentong Chen, Xi Chen, Xi Chen, Xiang Chen, Xiaojun Chen, Xie Chen, Xinyue Chen, Xinyue Chen, Xinyue Chen, Xu Chen, Xuanang Chen, Yang Chen, Yang Chen, Yankai Chen, Yanquan Chen, Yanran Chen, Yi Chen, Yi-pei Chen, Yi-syuan Chen, Yifan Chen, Yige Chen, Yijun Chen, Yilong Chen, Yin-jen Chen, Yingfa Chen, Yiqun Chen, Yiran Chen, Yiwen Chen, Yixin Chen, Yiyi Chen, Yongrui Chen, Yuhan Chen, Yuheng Chen, Yulin Chen, Yun Chen, Yuxuan Chen, Yuyan Chen, Zhanpeng Chen, Zhengyu Chen, Zhi-yuan Chen, Zhihong Chen, Zhipeng Chen, Zhongwu Chen, Zhumin Chen, Zhuowei Chen, Ziru Chen, Zixin Chen, Ziyang Chen, Zulong Chen, Daixuan Cheng, Dawei Cheng, Fan Cheng, Gang Cheng, Hong Cheng, Jiale Cheng, Jian Cheng, Jianpeng Cheng, Jiayang Cheng, Jingwei Cheng, Julius Cheng, Kewei Cheng, Liang Cheng, Luyao Cheng, Minhao Cheng, Myra Cheng, Pengzhou Cheng, Qinyuan Cheng, Reynold Cheng, Sijie Cheng, Sitao Cheng, Siyuan Cheng, Wei Cheng, Xianfu Cheng, Xiaoxia Cheng, Xiaoxue Cheng, Xin Cheng, Xinle Cheng, Xize Cheng, Yao Cheng, Yi Cheng, Zehua Cheng, Zifeng Cheng, Liu Chengwu, Lin Lee Cheong, Yun-gyung Cheong, Emmanuele Chersoni, Rendi Chevi, Oscar Chew, Anshuman Chhabra, Pohan Chi, Yew Ken Chia, Cheng-han Chiang, Ting-rui Chiang, Si-chi Chin, Abhinav Chinta, Patricia Chiril, Luis Chiruzzo, Elena Chistova, Pranjal A Chitale, Nima Chitsazan, Ming-chang Chiu, Yu Ying Chiu, Hyundong Justin Cho, Hyunsouk Cho, Ikhyun Cho, Ji Yong Cho, Kyunghyun Cho, Myeongah Cho, Sukmin Cho, Won Ik Cho, Young Min Cho, Rochelle Choenni, Donghee Choi, Eunkyung Choi, Hongseok Choi, Jaesik Choi, Jason Ingyu Choi, Joon-young Choi, Juhwan Choi, Jungwook Choi, Minseok Choi, Sehyun Choi, Seungtaek Choi, Sungchul Choi, Yongsuk Choi, Yubin Choi, Mukund Choudhary, Monojit Choudhury, Sagnik Ray Choudhury, Katsuki Chousa, Yinlam Chow, Arijit Ghosh Chowdhury, Somnath Basu Roy Chowdhury, Tanya Chowdhury, George Chrysostomou, Minh Duc Chu, Xiaowen Chu, Zheng Chu, Tat-seng Chua, Yu-neng Chuang, Yun-shiuan Chuang, Yung-sung Chuang, Ekapol Chuangsuwanich, Chanyoung Chung, Hyunseung Chung, John Joon Young Chung, Abu Nowshed Chy, Philipp Cimiano, Jorge Civera, Christian Clark, Christopher Clark, Elizabeth Clark, Christopher Clarke, Chloé Clavel, Stéphane Clinchant, Miruna Adriana Clinciu, Maximin Coavoux, Alexandru Coca, Oana Cocarascu, Amir David Nissan Cohen, Daniel Cohen, Nachshon Cohen, Vanya Cohen, Vincent Cohen-addad, Luisa Coheur, Trevor Cohn, Eliana Colunga, Prakash Mandayam Comar, Tianshuo Cong, Xin Cong, Mathieu Constant, Fierro Constanza, Danish Contractor, Owen Cook, Rodolfo Corona, Luciano Del Corro, Adrian Cosma, Luis Morgado Da Costa, Marta R. Costa-

jussà, Adriana Valentina Costache, Ryan Cotterell, Mickael Coustaty, Alan Cowap, Florinel Alin Croitoru, Huimin Cui, Jiajia Cui, Jin Cui, Qu Cui, Shaobo Cui, Shiyao Cui, Wanyun Cui, Wendi Cui, Yuanning Cui, Washington Cunha, Anna Currey, Alba Cercas Curry, Amanda Cercas Curry, Tamara Czinczoll, Erion Çano

Mathieu D'aquin, Jennifer D'souza, Gautier Dagan, Ido Dagan, Nico Daheim, Deborah A. Dahl, Daniel Dahlmeier, Bin Dai, Hong-jie Dai, Hongliang Dai, Huangyu Dai, Quanyu Dai, Shih-chieh Dai, Sunhao Dai, Xiang Dai, Yong Dai, Parag Pravin Dakle, Dwip Dalal, David Dale, Hercules Dalianis, Majid Daliri, Elizabeth M. Daly, Greta Damo, Soham Dan, Yuhao Dan, Ankit Dangi, Mai Hoang Dao, Zhang Daoan, Guy Dar, Anirban Das, Dipankar Das, Sarkar Snigdha Sarathi Das, Souvik Das, Mihai Dascalu, Debajyoti Dasgupta, Sourish Dasgupta, Tirthankar Dasgupta, Saloni Dash, Do Huu Dat, Debtanu Datta, Sam Davidson, Brian Davis, Christopher Davis, Hillary Dawkins, Min-yuh Day, Jiang Dazhi, Nicholas Deas, Alok Debnath, Hiroyuki Deguchi, Derrick Goh Xin Deik, Jean-benoit Delbrouck, Gianluca Demartini, Çağatay Demiralp, Dina Demner-fushman, Steve Deneefe, Ailin Deng, Cheng Deng, Chunyuan Deng, Guanzhi Deng, Hexuan Deng, Jiawen Deng, Jingcheng Deng, Naihao Deng, Shuiguang Deng, Shuwen Deng, Yue Deng, Zhenyun Deng, Zheye Deng, Zhijie Deng, Pavel Denisov, Omer Faruk Deniz, Sourabh Dattatray Deoghare, Al Depope, Jan Milan Deriu, Franck Dernoncourt, Aniket Deroy, Akshar Prabhu Desai, Meera Desai, Tejash Desai, Maunendra Desarkar, Vijeta Deshpande, Nina Dethlefs, Chaitanya Devaguptapu, Joseph Dexter, Suvodip Dey, Tejas Dhamecha, Prajit Dhar, Ruchira Dhar, Paramveer Dhillon, Aissatou Diallo, Xiaolei Diao, Xingjian Diao, Adji Bousso Dieng, Stefan Dietze, Barbara Dieugenio, Dimitar Iliyanov Dimitrov, Vania Dimitrova, Jacob Dineen, Bosheng Ding, Chuanghao Ding, Dujian Ding, Guiguang Ding, Hanxing Ding, Peng Ding, Shuangrui Ding, Shuoyang Ding, Wenjian Ding, Wenxuan Ding, Xiao Ding, Yifan Ding, Yifeng Ding, Yifu Ding, Yihao Ding, Ying Ding, Ying Ding, Yuyang Ding, Zifeng Ding, Zixiang Ding, Song Dingjie, Saket Dingliwal, Tu Anh Dinh, Tanvi Dinkar, Shubhashis Roy Dipta, Ajay Divakaran, Anuj Diwan, Amirbek Djanibekov, Heejin Do, Jaeyoung Do, Phong Nguyen-thuan Do, Susana Sotelo Docio, Sumanth Doddapaneni, Rama Doddipatla, Pierre Dognin, Kosuke Doi, Tobias Domhan, Gregor Donabauer, Bin Dong, Daize Dong, Guanting Dong, Hang Dong, Haoyu Dong, Jianfeng Dong, Junnan Dong, Kuicai Dong, Lu Dong, Ming Dong, Mingwen Dong, Peijie Dong, Rui Dong, Xiangjue Dong, Yijiang River Dong, Yuhao Dong, Yushun Dong, Zican Dong, Bonaventure F. P. Dossou, Longxu Dou, Qingyun Dou, Yao Dou, Antoine Doucet, Mohammad Javad Dousti, Mauro Dragoni, Mark Dras, Mark Dredze, Felix Drinkall, Rotem Dror, Dayou Du, Gaoyuan Du, Haowei Du, Jusen Du, Minxin Du, Weihong Du, Wenjie Du, Wenyu Du, Xiaoyong Du, Yali Du, Yanrui Du, Yifan Du, Yiming Du, Yupei Du, Haodong Duan, Jiaxin Duan, Jinhao Duan, Jiuding Duan, Juanyong Duan, Junwen Duan, Lixin Duan, Shaohua Duan, Sufeng Duan, Zenghao Duan, Zhichao Duan, Haim Dubossarsky, Fanny Ducel, George Dueñas, Richard Dufour, Lavinia Dunagan, Ewan Dunbar, Benjamin Van Durme, Ritam Dutt, Subhabrata Dutta, Jennifer Dy, Nils Dycke, Esra Dönmez

Haihong E, Oliver Eberle, Seth Ebner, Jessica Maria Echterhoff, Ali Edalat, Ali Edalati, Ghebriout Mohamed Imed Eddine, Lilach Eden, Carl Edwards, Paul Van Eecke, Steffen Eger, Koji Eguchi, Yo Ehara, Maud Ehrmann, Bryan Eikema, Liat Ein-dor, Alon Eirew, Roald Eiselen, Mohamed Elaraby, Passant Elchafei, Hoda Eldardiry, Mohamed Elgaar, T. Mark Ellison, Muhammad Nael Elnokrashy, Tamer Elsayed, Mai Elsherief, Micha Elsner, Yahya Emara, Matthew M. Engelhard, Sugyeong Eo, Elena V. Epure, Pierre Erbacher, Maksim Eremeev, Justus-jonas Christoph Erker, Brendan Raffel Evers, Chandra Kiran Reddy Evuru, Ana Ezquerro

Marzieh Fadaee, Fartash Faghri, Md Fahim, Neele Falk, Boi Faltings, Jiabin Fan, Rui Fan, Shengda Fan, Siqi Fan, Wenqi Fan, Xiran Fan, Yao-chung Fan, Yongqi Fan, Yue Fan, Zhaoxin Fan, Zhiyuan Fan, Chunrong Fang, Feiteng Fang, Guian Fang, Haishuo Fang, Jiabao Fang, Jinyuan Fang, Ming Fang, Minghong Fang, Minghui Fang, Qingkai Fang, Qixiang Fang, Tao Fang, Tianqing

Fang, Wei Fang, Xinyu Fang, Yi Fang, Yin Fang, Yuan Fang, Yuejian Fang, Zhaohao Fang, Zhixuan Fang, Youmna Farag, Reza Farahbakhsh, Gonçalo Rui Faria, Delia Irazu Hernandez Farias, António Farinhas, Richard Farkas, Sakina Fatima, Pedro Faustini, Mohsen Fayyaz, Arnisa Fazla, Pooyan Fazli, Amir Feder, Fabio James Fehr, Yu Fei, Zhaoye Fei, Nils Feldhus, Sergey Feldman, Andrew Zhuoer Feng, Dongji Feng, Guhao Feng, Huawen Feng, Lei Feng, Tao Feng, Weixi Feng, Xiaoyue Feng, Xincan Feng, Yang Feng, Yansong Feng, Yi Feng, Yu Feng, Yubo Feng, Yujie Feng, Zihao Feng, Zijian Feng, Pan Fengjun, Alena Fenogenova, Teddy Ferdinan, Martin Ferianc, Libby Ferland, Jared Fernandez, Nigel Fernandez, Aloka Fernando, Thomas Palmeira Ferraz, Sheridan Feucht, Julian Fierrez, Elena Filatova, James Darrell Finch, Sarah E. Finch, Matthew Finlayson, Hamed Firooz, Asja Fischer, Jillian Fisher, Jack Fitzgerald, Margaret Fleck, Eve Fleisig, James Flemings, Lorenzo Jaime Yu Flores, José A.r. Fonollosa, Marcio Fonseca, Negar Foroutan, Karën Fort, Robert Frank, Stella Frank, Alexander M. Fraser, Diego Frassinelli, Dayne Freitag, Simon Frieder, Giacomo Frisoni, Markus Frohmann, Arvid Frydenlund, Biao Fu, Dayuan Fu, Harvey Yiyun Fu, Jinlan Fu, Peng Fu, Tingchen Fu, Xiyan Fu, Yicheng Fu, Yun Fu, Zichuan Fu, Zuohui Fu, Dennis Fucci, Yoshinari Fujinuma, Fumiyo Fukumoto, Kotaro Funakoshi, Hiroaki Funayama, Shaz Furniturewala, Hiroki Furuta

Matteo Gabburo, Guillaume Gadek, Gaelguibon, Baban Gain, Robert J. Gaizauskas, Ya'akov Gal, Jay Gala, Alessio Galatolo, Marcos Martínez Galindo, Lukas Paul Achatius Galke, Martina Galletti, Jack Gallifant, Aram Galstyan, Pablo Gamallo, Andrew Gambardella, Yanglei Gan, Yujian Gan, Sriram Ganapathy, Mona Gandhi, Adithya V Ganesan, Balaji Ganesan, Debasis Ganguly, William Gantt, Tanuja Ganu, Bofei Gao, Changjiang Gao, Chao Gao, Chen Gao, Chufan Gao, Cuiyun Gao, Dehong Gao, Difei Gao, Ge Gao, Hongcheng Gao, Jiahui Gao, Jing Gao, Jinglong Gao, Jingsheng Gao, Jinhua Gao, Jun Gao, Jun Gao, Lang Gao, Mingqi Gao, Muhan Gao, Neng Gao, Pengzhi Gao, Qiang Gao, Qiyue Gao, Rena Wei Gao, Rujun Gao, Shangqian Gao, Shen Gao, Shengxiang Gao, Songyang Gao, Wei Gao, Xiang Gao, Xiaoxue Gao, Yingbo Gao, Yisen Gao, Zhujin Gao, Claire Gardent, Ankur Garg, Shweta Garg, Joseph Gatto, Himaanshu Gauba, Eric Gaussier, Vagrant Gautam, Jiaxin Ge, Mengshi Ge, Shiping Ge, Xiou Ge, Yubin Ge, Tom Gedeon, Michaela Geierhos, Robert Geislinger, Aryo Pradipta Gema, Debela Tesfaye Gemechu, Josef Van Genabith, Barbara Gendron, Ruiying Geng, Saibo Geng, Xiang Geng, Yuxia Geng, Enfa Rose George, Ariel Gera, Walter Gerych, Sadaf Ghaffari, Abdul Ghafoor, Omid Ghahroodi, Karim Ghonim, Kripabandhu Ghosh, Sayan Ghosh, Shaona Ghosh, Soumitra Ghosh, Sreyan Ghosh, Michael Ginn, Luca Gioffrè, Leander Girrbach, Raja Giryes, Adrià De Gispert, Fausto Giunchiglia, Michael Glass, Anna Glazkova, Kristina Gligoric, Max Glockner, Ameya Godbole, Anmol Goel, Arnav Goel, Evangelia Gogoulou, Sim Kuan Goh, Usman Gohar, Marcel Gohsen, Yoav Goldberg, Jonas Golde, Tomas Goldsack, Shyamnath Gollakota, Sujatha Das Gollapalli, Michal Golovanevsky, Juan Pablo Zuluaga Gomez, Marcos A Goncalves, Elizaveta Goncharova, Chen Gong, Chen Gong, Cheng Gong, Jiaying Gong, Kaixiong Gong, Neil Gong, Shansan Gong, Yihong Gong, Zhiguo Gong, Ziwei Gong, Aitor Gonzalez-agirre, Julio Gonzalo, Deepak Gopinath, Dhiman Goswami, Koustava Goswami, Sujata Goswami, Boyu Gou, Yunhao Gou, Antoine Gourru, Venkata Subrahmanyan Govindarajan, Pawan Goyal, Jorge Gracia, Carine Graff, Michael Granitzer, Christan Grant, Candida Maria Greco, Tommaso Green, Derek Greene, Rachel Greenstadt, Matt Grenander, Thomas L. Griffiths, Charlie Grimshaw, Milan Gritta, Georg Groh, Dagmar Gromann, Paul Groth, Bowen Gu, Hengrui Gu, Jia-chen Gu, Jinghang Gu, Naibin Gu, Shuhao Gu, Tiancheng Gu, Tianle Gu, Tianlong Gu, Xiangming Gu, Yi Gu, Yu Gu, Yuxian Gu, Yuxuan Gu, Zhouhong Gu, Zijin Gu, Eleonora Gualdoni, Jian Guan, Nan Guan, Quanlong Guan, Shuhao Guan, Xin Guan, Xinyan Guan, Yong Guan, Li Hong Guang, Reto Gubelmann, Bhanu Prakash Reddy Guda, Imane Guellil, Marco Guerini, Michael Guerzhoy, Dipanwita Guhathakurta, Mustafa Omer Gul, Varun Gumma, Tom Gunter, Boyang Guo, Fenfei Guo, Han Guo, Hangyu Guo, Hongcheng Guo, Jia Guo, Jiaxin Guo, Jinyu Guo, Kan Guo, Kehan Guo, Lingbing Guo, Meiqi Guo, Qingpei Guo, Qipeng Guo, Ruohao Guo, Shangwei Guo, Shaoru Guo, Shasha Guo,

Shoutao Guo, Siyi Guo, Siyuan Guo, Taicheng Guo, Wenya Guo, Xiaobo Guo, Xinnan Guo, Xu Guo, Yanzhu Guo, Yike Guo, Yue Guo, Yuhang Guo, Zhen Guo, Zhicheng Guo, Zhiyu Guo, Ziyu Guo, Akash Gupta, Ankita Gupta, Ashim Gupta, Deepak Gupta, Gopal Gupta, Jai Gupta, Lavanya Gupta, Manish Gupta, Sameer Gupta, Sonam Gupta, Umang Gupta, Vipul Gupta, Timon Gurcke, Iryna Gurevych, V Gurucharan, Sireesh Gururaja, Yoan Gutierrez, Ximena Gutierrez-vasques, Daehoon Gwak, Jeremy Gwinnup

Thanh-le Ha, Shohreh Haddadan, Nikita Haduong, Samar Haider, Mao Haitao, Jan Hajič, Huda Hakami, Ahmad Dawar Hakimi, Ido Hakimi, Sherzod Hakimov, Scott A. Hale, Karina H Halevy, Coleman Haley, Patrick Haller, Skyler Hallinan, Hassan Hamad, Karen Hambardzumyan, Ayoub Hammal, Boran Han, Caren Han, Cheng Han, Chengcheng Han, Donghoon Han, Dou Shi Han, Haoyu Han, Hojae Han, Janghoon Han, Jiale Han, Jiuzhou Han, Kanyao Han, Lifeng Han, Meng Han, Namgi Han, Seungju Han, Shijie Han, Simeng Han, Wei Han, William Han, Xiaoqi Han, Xu Han, Xu Han, Xudong Han, Xue Han, Yang Han, Yo-sub Han, Ziwen Han, Adeep Hande, Zhang Hanlin, Michael Hanna, Chuzhan Hao, Hailin Hao, Jianye Hao, Qirui Hao, Tianyong Hao, Xiaoyang Hao, Yaru Hao, Yongchang Hao, Wang Haotian, Rejwanul Haque, Romain Harang, David Harbecke, Karin Harbusch, Momchil Hardalov, Fabrice Y Harel-canada, Stefan Harmeling, Ian Harris, Thomas Hartvigsen, Md Rakibul Hasan, Md. Arid Hasan, Mohammad Saqib Hasan, Maram Hasanain, Taku Hasegawa, Mark A. Hasegawa-johnson, Kazuma Hashimoto, Faegheh Hasibi, Islam A. Hassan, Michael Hassid, Nabil Hathout, Hans Ole Hatzel, Shreya Havaldar, Akio Hayakawa, Katsuhiko Hayashi, Amir Hazem, Rima Hazra, Bailan He, Bingsheng He, Bingxiang He, Bolei He, Bowei He, Chaoqun He, Conghui He, Estrid He, Guoxiu He, Hangfeng He, Jianfeng He, Jiangen He, Junda He, Junqing He, Kai He, Kun He, Liqi He, Liyang He, Pengfei He, Pengfei He, Pinjia He, Qianyu He, Qing He, Ruifang He, Shwai He, Tao He, Tingting He, Xiangjian He, Xiangnan He, Xingwei He, Xinyi He, Xuehai He, Xuming He, Yifei He, Zexue He, Zhaofeng He, Zhitao He, Zihao He, Jennifer Healey, Larry Heck, Tobias Hecking, Johannes Heinecke, Maximilian Heinrich, Benjamin Heinzerling, Jindřich Helcl, William Barr Held, Oliver Hellwig, Chadi Helwe, Bhathiya Hemanthage, James Henderson, Iris Hendrickx, Amey Hengle, Aron Henriksson, Christian Herold, Jack Hessel, Anthony Hills, Tatsuya Hiraoka, Eran Hirsch, Julia Hirschberg, Youssef Hmamouche, Namgyu Ho, Xanh Ho, Nhat Minh Hoang, Thai Quoc Hoang, Thanh Tung Hoang, Vu Cong Duy Hoang, Sanne Hoeken, David Crossland Hogg, Carolin Holtermann, Ukyo Honda, Vlad Hondru, Giwon Hong, Jiwoo Hong, Jixiang Hong, Pengfei Hong, Ruixin Hong, Seokhee Hong, Yihuai Hong, Yu Hong, Yuan Hong, Zijin Hong, Hongbinzhanghitsz, Liu Hongjun, Jia Hongrui, Or Honovich, Anthony Hoogs, Zachary William Hopton, Mohammed Moshiul Hoque, Andrea Horbach, Zachary Horvitz, Sho Hoshino, Faezeh Hosseini, Peyman Hosseini, Veronique Hoste, Abe Bohan Hou, Bairu Hou, Benjamin Hou, Guiyang Hou, Lu Hou, Rui Hou, Wenjun Hou, Yifan Hou, Yingyan Hou, Yu Hou, Zejiang Hou, Zhenyu Hou, Phillip Howard, David M Howcroft, Nick Howell, Chi-yuan Hsiao, I-hung Hsu, Wynne Hsu, Yili Hsu, Phu-mon Htut, Baotian Hu, Bodun Hu, Brian H Hu, Chenxi Hu, Guimin Hu, Hai Hu, Haifeng Hu, Hanxu Hu, He Hu, Hongxin Hu, Jinpeng Hu, Jinyi Hu, Jun Hu, Lianyu Hu, Lijie Hu, Mengkang Hu, Mengzhou Hu, Minda Hu, Minghao Hu, Nan Hu, Qi Hu, Qisheng Hu, Renfen Hu, Renjun Hu, Songbo Hu, Tiancheng Hu, Vincent Tao Hu, Wenbin Hu, Wenbo Hu, Wenxiang Hu, Xiao Hu, Xiaomeng Hu, Xiaoyu Hu, Xinshuo Hu, Xinyu Hu, Xiyang Hu, Yebowen Hu, Yiran Hu, Yuchen Hu, Yutong Hu, Zhe Hu, Zhengmian Hu, Zhengyu Hu, Zhiyuan Hu, Zi-yuan Hu, Hang Hua, Ting Hua, Wenyue Hua, Yuncheng Hua, Baizhou Huang, Chao-wei Huang, Chengkai Huang, Chengsong Huang, Chengyu Huang, Chengyue Huang, Chenyang Huang, Chu-ren Huang, Chun-ying Huang, Dong Huang, Fan Huang, Feiran Huang, Hanxian Huang, Hao Huang, Haojing Huang, Haoyu Huang, Hen-hsen Huang, Heng Huang, Hongtao Huang, Hsiu-yuan Huang, Hui Huang, Hujiang Huang, Jen-tse Huang, Jerry Huang, Jianheng Huang, Jimin Huang, Jin-xia Huang, Jinfa Huang, Jingyuan Huang, Junwei Huang, Kaiyu Huang, Kaizhu Huang, Kun Huang, Langlin Huang, Lei Huang, Li Huang, Liusheng Huang, Minghui Huang, Minlie Huang, Peijie Huang, Pengcheng Huang,

Qingbao Huang, Qiushi Huang, Quzhe Huang, Rikui Huang, Ruixuan Huang, Ruizhang Huang, Shijia Huang, Shizhou Huang, Shujian Huang, Shuo Huang, Siyuan Huang, Wei Huang, Weijing Huang, Wenhao Huang, Wenxin Huang, Xiang Huang, Xiaolei Huang, Yan Huang, Yi Huang, Yin Jou Huang, Yinxuan Huang, Yongfeng Huang, Youcheng Huang, Yu-shiang Huang, Yue Huang, Yuhao Huang, Yukun Huang, Zanming Huang, Zeyu Huang, Zhen Huang, Zhichao Huang, Zhongzhan Huang, Zixian Huang, Yuanxiang Huangfu, Zhang Huaping, Wang Huazheng, Patrick Huber, Frederikus Hudi, Mathew John Huerta-enochian, Bo Hui, Zheng Hui, Rudali Huidrom, Huiyingcao, Huiyingcao, Lawrence Hunter, Shuning Huo, Maeve Hutchinson, Nguyen Thanh Huy, Dae Yon Hwang, Eunjeong Hwang, Jena D. Hwang, Seonjeong Hwang, Truong Son Hy, Katharina Hämmerl, Mika Hämäläinen, Anna Hätty

Mahardika Krisna Ihsani, Hal Daumé Iii, Taichi Iki, Nikolai Ilinykh, Dmitry Ilvovsky, Solee Im, Ayyoob Imani, Joseph Marvin Imperial, Muhammad Imran, Michimasa Inaba, Mert Inan, Go Inoue, Ioan-bogdan Iordache, David Martinez Iraola, Ozan Irsoy, Yoichi Ishibashi, Tatsuya Ishigaki, Alvi Md Ishmam, Md Asiful Islam, Md Rakibul Islam, Md Saiful Islam, Mohammed Saidul Islam, Mete Ismayilzada, Takumi Ito, Alexei V. Ivanov, Vladimir Ivanov, Hamish Ivison, Yusuke Iwasawa, Bhavani Iyer, Rishabh Iyer, Gautier Izacard

Karthika N J, Manoj Balaji J, Musashi Jacobs-harukawa, Aditya Ajay Jadhav, Aishwarya Jadhav, Aref Jafari, Akshay Jagatap, Arushi Jain, Rishabh Jain, Sanil Jain, Sparsh Jain, Yash Jain, Mimansa Jaiswal, Terne Sasha Thorn Jakobsen, Shruti Jalan, Surgan Jandial, Dongjun Jang, Eugene Jang, Jihyoung Jang, Joonwon Jang, Myeongjun Jang, Seongbo Jang, Young Kyun Jang, Yunah Jang, Peter Jansen, Arkadiusz Janz, Amit Jaspal, Sujay Kumar Jauhar, Dávid Javorský, Sai Muralidhar Jayanthi, Nicolaas Paul Jedema, Christopher William Jenkins, Eojin Jeon, Hyewon Jeon, Yejin Jeon, Minbyul Jeong, Soyeong Jeong, Sullam Jeoung, Bin Ji, Donghong Ji, Heng Ji, Jiayi Ji, Ke Ji, Shengpeng Ji, Tianbo Ji, Wei Ji, Yixin Ji, Yuelyu Ji, Zhenzhou Ji, Ziwei Ji, Zongcheng Ji, Furong Jia, Jinghan Jia, Pengyue Jia, Qi Jia, Ruipeng Jia, Xueying Jia, Yuxiang Jia, Liu Jiaheng, Ping Jian, Yiren Jian, Zhongquan Jian, Chao Ya Jiang, Chunyang Jiang, Dongfu Jiang, Dongwei Jiang, Fan Jiang, Fengqing Jiang, Gongyao Jiang, Guochao Jiang, Haiyun Jiang, Jinhao Jiang, Jiyue Jiang, Junfeng Jiang, Junzhe Jiang, Jyun-yu Jiang, Lai Jiang, Lan Jiang, Liwei Jiang, Meng Jiang, Ming Jiang, Nan Jiang, Nan Jiang, Ning Jiang, Qian Jiang, Shan Jiang, Tianyu Jiang, Tintin Jiang, Wenyuan Jiang, Xiaotong Jiang, Yichen Jiang, Yidi Jiang, Yuxin Jiang, Zhiwei Jiang, Zhongtao Jiang, Zhuoren Jiang, Ziyan Jiang, Ziyue Jiang, Zui Jiang, Jianghoucheng, Fangkai Jiao, Jian Jiao, Wenpin Jiao, Wenxiang Jiao, Yizhu Jiao, Bowen Jin, Chuanyang Jin, Guoqing Jin, Hai Jin, Han Jin, Hongye Jin, Jiho Jin, Junfeng Jin, Kyohoon Jin, Li Jin, Lianwen Jin, Lisa Jin, Mali Jin, Mingyu Jin, Peiquan Jin, Renren Jin, Rize Jin, Ruochen Jin, Ruochun Jin, Souyoung Jin, Tao Jin, Xiaolong Jin, Xiaomeng Jin, Xisen Jin, Yaohui Jin, Yeying Jin, Yiqiao Jin, Zhe Jin, Zhi Jin, Zhuoran Jin, Disha Jindal, Ishan Jindal, Liping Jing, Liqiang Jing, Wu Jing, Yuu Jinnai, Hwiyeol Jo, Yongrae Jo, Moa Johansson, Neha Anna John, Kristen Johnson, Jiahuei Joo, Alipio Jorge, Joemon M. Jose, Abhinav Joshi, Aditya Joshi, Anant Joshi, Brihi Joshi, Harshit Joshi, Rishabh Joshi, Mingxuan Ju, Tianjie Ju, Xincheng Ju, Yiming Ju, Gurusha Juneja, Changsoo Jung, Dahyun Jung, Dongwon Jung, Haeji Jung, Sangkeun Jung, Woohwan Jung, Yeonjoon Jung, He Junwei, Lu Junyu, Juraj Juraska, Nika Jurov, Tom S. Juzek

Karthikeyan K, Mohsinul Kabir, Anubha Kabra, David Kaczér, Ram Mohan Rao Kadiyala, Marek Kadlčík, Kazuma Kadowaki, Patrick Kahardipraja, Tomasz Jan Kajdanowicz, Abisek Rajakumar Kalarani, Jan-christoph Kalo, Nishant Kambhatla, Hirotaka Kameko, Sabyasachi Kamila, Ryo Kamoi, Jonathan Kamp, Onno P. Kampman, Jaap Kamps, Mahammed Kamruzzaman, Hiroshi Kanayama, Daniel Kang, Deokhyung Kang, Dongjin Kang, Gi-cheon Kang, Jaewoo Kang, Jaewook Kang, Junmo Kang, Liu Ze Kang, Minki Kang, Pilsung Kang, Xiaoxi Kang, Zhao Kang, Yoshinobu Kano, Hung-yu Kao, Rajvi Kapadia, Pinar Karagoz, Antonia Karamolegkou, Mladen Karan,

Hima Karanam, Amir Hossein Kargaran, Priyanka Kargupta, Akbar Karimi, Sanjeev Kumar Karn, Zohar Karnin, Marzena Karpinska, Fakhri Karray, Pradeep Karuturi, Pranav Kasela, Gautam Siddharth Kashyap, Gjergji Kasneci, Surya Kasturi, Tushar Kataria, Shunya Kato, Yoshihide Kato, Gregorios A Katsios, Ayush Kaushal, Muhammad Dehan Al Kautsar, Parsa Kavehzadeh, Pride Kavumba, Daisuke Kawahara, Masayuki Kawarada, Yoshifumi Kawasaki, Roman Kazakov, Hideto Kazawa, Ashkan Kazemi, Seyed Mehran Kazemi, Wenjun Ke, Xiaolan Ke, Sedrick Scott Keh, Mamadou K. Keita, Katherine A. Keith, Simon Keizer, Amr Keleg, Mikaela Keller, Orgad Keller, Casey Kennington, Roman Kern, Natthawut Kertkeidkachorn, Vaishakh Keshava, Lee Kezar, Baber Khalid, Muhammad Khalid, Osama Khalid, Mohammed Khalilia, Aleem Khan, Latifur Khan, Mohammed Safi Ur Rahman Khan, Salman Khan, Sujay Khandagale, Vivek Khetan, Zahra Khodagholi, Md Tawkat Islam Khondaker, Ashiqur R. Khudabukhsh, Urja Khurana, Varun Khurana, Dayeon Ki, Johannes Kiesel, Zlata Kikteva, Jihyung Kil, Halil Kilicoglu, Bugeun Kim, Byeonggeun Kim, Chae Won Kim, Chaeeun Kim, Changhun Kim, Dahyun Kim, Dongwoo Kim, Eunsu Kim, Geewook Kim, Gene Louis Kim, Gunhee Kim, Gyeongman Kim, Gyuwan Kim, Hansaem Kim, Harksoo Kim, Hazel H. Kim, Heeseung Kim, Hongjin Kim, Hwichan Kim, Hyounghun Kim, Hyuhng Joon Kim, Jaehyung Kim, Jaein Kim, Jaekyeom Kim, Jaeyoung Kim, Jeonghoon Kim, Jeonghwan Kim, Jieyong Kim, Jiho Kim, Jihyuk Kim, Jinsung Kim, Jiseon Kim, Jongho Kim, Jongyoon Kim, Jung-jae Kim, Junho Kim, Junyeob Kim, Junyeong Kim, Kang-min Kim, Kangil Kim, Keonwoo Kim, Kon Woo Kim, Michelle Kim, Minbeom Kim, Minju Kim, Minseok Kim, Minsoo Kim, Minsoo Kim, Minsu Kim, Misoo Kim, Misuk Kim, Seungone Kim, Sun Kim, Sungchul Kim, Sungdong Kim, Taehwan Kim, Takyoung Kim, Yeeun Kim, Yongil Kim, Youna Kim, Youngbin Kim, Youngkyoung Kim, Yujin Kim, Yumin Kim, Zae Myung Kim, Brendan King, Tracy Holloway King, Frederic Kirstein, Chunyu Kit, Diego Klabjan, Christopher Klamm, Jens Kleesiek, Đorđe Klisura, Marius Kloft, Marianne De Heer Kloots, Kate Knill, Dayoon Ko, Jongwoo Ko, Sang-ki Ko, Youngjoong Ko, Goro Kobayashi, Ichiro Kobayashi, Sosuke Kobayashi, Elena Kochkina, Prashant Kodali, Jordan Kodner, Hyukhun Koh, Ryosuke Kohita, Guneet Singh Kohli, Ryuto Koike, Stanley Kok, Jachym Kolar, Olga Kolesnikova, Prateek Kolhar, Alexander Koller, Sai Koneru, Cunliang Kong, Fang Kong, Fanshuang Kong, Jun Kong, Keyi Kong, Linghe Kong, Lingxing Kong, Tao Kong, Myoung-wan Koo, Seonmin Koo, Maarit Koponen, Yeganeh Kordi, Katerina Korre, Roman Koshkin, Katsunori Kotani, Neema Kotonya, Ziyi Kou, Mahnaz Koupaee, George Kour, Punit Singh Koura, Geza Kovacs, Adriana Kovashka, Lea Krause, Robert Kraut, Markus Kreuzthaler, Satyapriya Krishna, Adit Krishnan, Nikhil Krishnaswamy, Canasai Kruengkrai, Joni Kruijsbergen, Lun-wei Ku, Max Ku, Jiayi Kuang, Jun Kuang, Marek Kubis, Keito Kudo, Sachit Kuhar, Adithya Kulkarni, Atharva Kulkarni, Atharva Kulkarni, Mayank Kulkarni, Anoop Kumar, Bhargava Kumar, Deepak Kumar, Karun Kumar, Nalin Kumar, Nitesh Kumar, Rishabh Kumar, Sachin Kumar, Sandeep Kumar, Shanu Kumar, Shashi Kumar, Sonal Kumar, Sujay S Kumar, Vishwajeet Kumar, Vivek Kumar, Yaman Kumar, Tharindu Sandaruwan Kumarage, Sadhana Kumaravel, Gitanjali Kumari, Tuhin Kundu, Po-nien Kung, Jenny Kunz, Olli Kuparinen, Yury Kuratov, Murathan Kurfalı, Kemal Kurniawan, Guy Kushilevitz, Mucahid Kutlu, Gleb Kuzmin, Andrey Kuznetsov, Ilia Kuznetsov, Kristian Kuznetsov, Alice Saebom Kwak, Beongwoo Kwak, Sunjun Kweon, Wonbin Kweon, Deuksin Kwon, Jingun Kwon, Se Jung Kwon, Taeyoon Kwon, Daeun Kyung, Abdullatif Köksal

Jorma Tapio Laaksonen, Gorka Labaka, Philippe Laban, Vincent Labatut, Cyril Labbe, Yanis Labrak, Tiziano Labruna, Oier Lopez De Lacalle, Avishek Lahiri, Allison Claire Lahnala, Eason Lai, Hanjiang Lai, Huiyuan Lai, Peichao Lai, Tuan Lai, Viet Dac Lai, Wen Lai, Yilong Lai, Yuhang Lai, Yuxuan Lai, Kushnareva Laida, Francesco Laiti, Rice Lake, Yair Lakretz, John P. Lalor, Albert Y.s. Lam, Jessica Lam, Hemank Lamba, Vasileios Lampos, Gerasimos Lampouras, Andre Lamurias, Disen Lan, Tian Lan, Zhenzhong Lan, Paul Landes, Hao Lang, Lukas Lange, Loic De Langhe, Karin De Langis, Philippe Langlais, David Langlois, Mateusz Lango, Mirella Lapata, Gabriella Lapesa, Martha Larson, Stefan Larson, Chaak-ming Lau, Jey Han Lau, Leo Laugier,

Ivano Lauriola, Leonard Lausen, Jack Laviolette, John Lawrence, Dung Duy Le, Duy Le, Hang Le, Hoang-quynh Le, Nghia T. Le, Nham Le, Thang Le, Kevin Leach, Joshua Ong Jun Leang, Gwénolé Lecorvé, Beomseok Lee, Bruce W Lee, Byung-jun Lee, Changhun Lee, Changki Lee, Changmin Lee, Cheryl Lee, Dohyeon Lee, Dong-ho Lee, Dongha Lee, Dongwon Lee, Fei-tzin Lee, Gary Lee, Gyeongeun Lee, Gyubok Lee, Hodong Lee, Hojae Lee, Hyunju Lee, Jaejin Lee, Jaeseong Lee, Janghwan Lee, Ji-ung Lee, Jimin Lee, Jinu Lee, Jiyoung Lee, John Sie Yuen Lee, Jongwuk Lee, Jung Hyun Lee, Jungseob Lee, Kangwook Lee, Katherine Lee, Kwonjoon Lee, Kyong-ho Lee, Lillian Lee, Mingyu Lee, Minhwa Lee, Nayeon Lee, Patrick Lee, Saehyung Lee, Sang-gil Lee, Sang-goo Lee, Sang-woo Lee, Sanwoo Lee, Seanie Lee, Seonghyeon Lee, Seungeon Lee, Seungyoon Lee, Sunbowen Lee, Sunkyung Lee, Suyeon Lee, Young-jun Lee, Younghun Lee, Youngwon Lee, Yukyung Lee, Fabrice Lefèvre, Joël Legrand, Fangyu Lei, Yibin Lei, Yuanyuan Lei, Zhenyu Lei, Markus Leippold, Christoph Leiter, Maxyn Leitner, Gaël Lejeune, Chak Tou Leong, Bruno Lepri, Pietro Lesci, Lauren Elizabeth Levine, Sergey Levine, Sharon Levy, William D. Lewis, Ang Li, Anqi Li, Bangzheng Li, Bing Li, Bo Li, Bobo Li, Bocheng Li, Bryan Li, Changhao Li, Changmao Li, Changqun Li, Changye Li, Chaozhuo Li, Chengkai Li, Chengming Li, Chengpeng Li, Chengxi Li, Chengzhengxu Li, Chengzu Li, Chong Li, Chuhan Li, Chunyang Li, Chuyuan Li, Cuiping Li, Daifeng Li, Dexun Li, Dingcheng Li, Diya Li, Dong Li, Dongsheng Li, Dongsheng Li, Dongyang Li, Dongyuan Li, Feng-lin Li, Ge Yi Li, Guozheng Li, Hang Li, Hao Li, Hao Li, Hao Li, Haochen Li, Haohang Li, Haonan Li, Haoran Li, Haoran Li, Haoxin Li, Haoyuan Li, Hongyu Li, Huan Li, Huayang Li, Hui Li, Hui Li, Huihan Li, Huiying Li, Jia Li, Jia Li, Jiaang Li, Jiahao Li, Jiahuan Li, Jialu Li, Jiang Li, Jianjun Li, Jianxin Li, Jiaoda Li, Jiaqi Li, Jiaxi Li, Jiaxuan Li, Jing Li, Jingxuan Li, Jinning Li, Jinyang Li, Jinyuan Li, Jixing Li, Juanzi Li, Juntao Li, Junxian Li, Junyi Li, Junzhuo Li, Kaixin Li, Kan Li, Ke Li, Kun Li, Kun Li, Kun Li, Lei Li, Li Li, Liang Li, Liang Li, Liangze Li, Linjing Li, Linjun Li, Linsen Li, Lishuang Li, Liyao Li, Lujun Li, Meng Li, Mengze Li, Miao Li, Min Li, Ming Li, Mingchen Li, Mingda Li, Mingjie Li, Mingming Li, Mingxiao Li, Mingzhe Li, Minzhi Li, Mukai Li, Na Li, Nian Li, Ning Li, Oliver Li, Peng Li, Ping Li, Qi Li, Qian Li, Qing Li, Qing Li, Qing Li, Qing Li, Qintong Li, Qiuchi Li, Ran Li, Raymond Li, Renhao Li, Rong Li, Rongsheng Li, Rui Li, Rui Li, Rui Li, Rui Li, Rui Li, Ruifan Li, Ruihao Li, Ruiqi Li, Ruixuan Li, Rumei Li, Ruosen Li, Sha Li, Shaobo Li, Shasha Li, Sheng Li, Sheng Li, Shengjie Li, Shichen Li, Shimin Li, Shiyao Li, Shuaiyi Li, Shuangyin Li, Shuo Li, Siyan Li, Sujian Li, Tianhao Li, Tianlin Li, Tianrui Li, Wei Li, Weijie Li, Wenda Li, Wenhao Li, Xia Li, Xiang Li, Xiang Li, Xiangci Li, Xiangtai Li, Xiangyang Li, Xianming Li, Xiao Li, Xiaochuan Li, Xiaomin Li, Xiaonan Li, Xiaopeng Li, Xiaopeng Li, Xiaoxuan Li, Xingxuan Li, Xinjin Li, Xintong Li, Xinze Li, Xinze Li, Xiujun Li, Xiuxing Li, Xuchen Li, Xuelong Li, Ya Li, Yafu Li, Yang Li, Yangning Li, Yanyang Li, Yawei Li, Yi Li, Yifei Li, Ying Li, Yinghao Li, Yingji Li, Yingya Li, Yinqiao Li, Yiqi Li, Yitong Li, Yitong Li, Yiwei Li, Yixia Li, Yixing Li, Yizhi Li, Yongqi Li, Yu Li, Yucheng Li, Yue Li, Yuepei Li, Yufei Li, Yuhan Li, Yunshui Li, Yunxin Li, Yunxuan Li, Yuqing Li, Zaijing Li, Zang Li, Zechen Li, Zehan Li, Zejun Li, Zekun Li, Zexin Li, Zhaonan Li, Zhaowei Li, Zhen Li, Zheng Li, Zheng Li, Zhenhao Li, Zhenwen Li, Zhixu Li, Zhiyu Li, Zhuohang Li, Zhuohao Li, Zichao Li, Zihan Li, Zihao Li, Zihao Li, Ziniu Li, Zizhong Li, Zongxi Li, Zongxia Li, Zongyao Li, Zuchao Li, Defu Lian, Jiachen Lian, Jianxun Lian, Zheng Lian, Chao-chun Liang, Chen Liang, Jenny Liang, Jiafeng Liang, Jiaqing Liang, Jinggui Liang, Junbang Liang, Sheng Liang, Shuailong Liang, Shuo Liang, Tian Liang, Tingting Liang, Wenxin Liang, Xiaobo Liang, Xiwen Liang, Xun Liang, Yingyu Liang, Yitao Liang, Yunlong Liang, Yupu Liang, Zhenwen Liang, Zihan Liang, Baohao Liao, Hao Liao, Huanxuan Liao, Jian Liao, Lizi Liao, Ming Liao, Minpeng Liao, Pei-jun Liao, Ruotong Liao, Weiheng Liao, Wenhui Liao, Xingming Liao, Yong Liao, Yusheng Liao, Zihan Liao, John Licato, Luca Di Liello, Wencke Liermann, Heuiseok Lim, Kyungtae Lim, Ser-nam Lim, Woohyung Lim, Woosang Lim, Tomasz Limisiewicz, Peerat Limkonchotiwat, Antonie Lin, Chi-heng Lin, Chu-cheng Lin, Chun Yi Lin, Guan-ting Lin, Hao-kun Lin, Haonan Lin, Hongzhan Lin, Huawei Lin, I-fan Lin, Inna Wanyin Lin, Jiayu Lin, Jieyu Lin, Lu Lin, Mingbao Lin, Minhua Lin, Pin-jie Lin, Qika Lin, Ruixi Lin, Run Lin, Tony Lin,

Weizhe Lin, Xueyuan Lin, Yen-ting Lin, Yi-cheng Lin, Ying-jia Lin, Youyuan Lin, Yueqian Lin, Yupian Lin, Zeqi Lin, Zhe Lin, Zhen Lin, Zheng Lin, Zhong Lin, Zhouhan Lin, Zihao Lin, Zongyu Lin, Matthias Lindemann, Peter Lindes, Chen Ling, Yuan Ling, Zhen-hua Ling, Zijian Ling, Tal Linzen, Gili Lior, Enrico Liscio, Johann-mattis List, Aiwei Liu, Alisa Liu, An Liu, Anthony Zhe Liu, Ben Liu, Bing Liu, Bingyan Liu, Bo Liu, Bo Liu, Bo Liu, Chao-lin Liu, Chaoqun Liu, Chen Cecilia Liu, Chengyuan Liu, Chuanyi Liu, Chunhua Liu, Cong Liu, Cong Liu, Daizong Liu, Danni Liu, Dongfang Liu, Dugang Liu, Emmy Liu, Enshu Liu, Fangchao Liu, Fei Liu, Fuxiao Liu, Gangao Liu, Gongshen Liu, Guangyi Liu, Guisheng Liu, Han Liu, Hang Liu, Hao Liu, Haotian Liu, Hongfu Liu, Hongyi Liu, Huadai Liu, Hui Liu, Hui Liu, Huijun Liu, Ji Liu, Jiacheng Liu, Jiamou Liu, Jianfeng Liu, Jiang Liu, Jiangming Liu, Jianxing Liu, Jiarui Liu, Jiateng Liu, Jiawei Liu, Jiejyun Liu, Jin Liu, Jin Liu, Jinghui Liu, Jingping Liu, Jingyao Liu, Juhua Liu, Jun Liu, Junlong Liu, Junpeng Liu, Junpeng Liu, Kang Liu, Kunpeng Liu, Lan Liu, Lihui Liu, Linfeng Liu, Ling Liu, Meng Liu, Ming Liu, Ming Liu, Minqian Liu, Nayu Liu, Nelson F. Liu, Ninghao Liu, Peipei Liu, Peiyu Liu, Pengyuan Liu, Ping Liu, Qiang Liu, Qiao Liu, Qijiong Liu, Qin Liu, Rui Liu, Rui Liu, Ruikang Liu, Ruixuan Liu, Ruiyang Liu, Ruoqi Liu, Sheng Liu, Shengjie Liu, Shengzhong Liu, Shicheng Liu, Shichun Liu, Shuai Liu, Shuheng Liu, Shujie Liu, Shukai Liu, Shuliang Liu, Shuqi Liu, Sichen Liu, Sijia Liu, Siyang Liu, Siyi Liu, Siyi Liu, Tengxiao Liu, Tianci Liu, Tianming Liu, Tianyang Liu, Tianyang Liu, Tianyu Liu, Ting Liu, Tingkai Liu, Tong Liu, Wei Liu, Wei Liu, Wei Liu, Weifeng Liu, Weihao Liu, Weize Liu, Wenhe Liu, Xiang Liu, Xiangrui Liu, Xiao Liu, Xiaoming Liu, Xiaoze Liu, Xin Liu, Xin Liu, Xinyue Liu, Xiping Liu, Xixian Liu, Xuebing Liu, Xueqing Liu, Xuye Liu, Yahui Liu, Yan Liu, Yanchen Liu, Yanchi Liu, Yang Liu, Yang Liu, Yang Aron Liu, Yanming Liu, Yantao Liu, Yanting Liu, Ye Liu, Ye Liu, Yi Liu, Yicun Liu, Yidi Liu, Yiding Liu, Yihong Liu, Ying Liu, Yinhong Liu, Yong Liu, Yong Liu, Yonghao Liu, Yongkang Liu, Yongmei Liu, Yu Lu Liu, Yuanchao Liu, Yuanxing Liu, Yuchen Liu, Yue Liu, Yuhan Liu, Yuhan Liu, Yuqing Liu, Yuxiang Liu, Yuxuan Liu, Zeming Liu, Zhengyuan Liu, Zhexiong Liu, Zheyuan Liu, Zhili Liu, Zhining Liu, Zhiwei Liu, Zhiyue Liu, Zhuo Liu, Zihan Liu, Zihang Liu, Zijie Liu, Zijing Liu, Zijun Liu, Zikang Liu, Ziqi Liu, Ziqiang Liu, Zitao Liu, Ziyi Liu, Zizhou Liu, Elena Lloret, Chun Hei Lo, Soda Marem Lo, Tien-hong Lo, Tyler Loakman, Ekaterina Lobacheva, Anamika Lochab, Andrea Loddo, Mengsay Loem, Rahul Lokesh, Erika Lombart, Dingkun Long, Do Xuan Long, Quanyu Long, Teng Long, Wanqiu Long, Xinwei Long, Yunfei Long, Michela Lorandi, Cedric Lothritz, Jian Lou, Qian Lou, Maxime Louis, Natalia Loukachevitch, Justin Lovelace, Bin Lu, Bo-ru Lu, Chaochao Lu, Heng-yang Lu, Hengtong Lu, Jianqiao Lu, Jinghui Lu, Jinliang Lu, Junru Lu, Lei Lu, Ning Lu, Peng Lu, Qingyu Lu, Qixi Lu, Sidi Lu, Wenpeng Lu, Xiaolei Lu, Xiaoxin Lu, Xin Lu, Xinyuan Lu, Yichao Lu, Yida Lu, Yining Lu, Yu Lu, Yu Lu, Yunfan Lu, Yuxing Lu, Yuxuan Lu, Zexin Lu, Zhiyong Lu, Nurul Lubis, Evan Lucas, Jason S Lucas, Bernd Ludwig, Riccardo Lunardi, Chu Fei Luo, Dongsheng Luo, Feng Luo, Fuwen Luo, Ge Luo, Guoqing Luo, Haoran Luo, Haozheng Luo, Hongyin Luo, Jiaming Luo, Jinghao Luo, Jinqi Luo, Junyu Luo, Man Luo, Minnan Luo, Ruilin Luo, Run Luo, Sichun Luo, Tianze Luo, Tongxu Luo, Weidi Luo, Xiang Luo, Xianzhen Luo, Xufang Luo, Yanchen Luo, Yang Luo, Ye Luo, Yingfeng Luo, Yiran Luo, Yong Luo, Yuanchang Luo, Yun Luo, Yuyu Luo, Zheheng Luo, Zhuang Luo, Ziyang Luo, Thang Luong, Mohammad Luqman, Lin Luyang, Bo Lv, Changze Lv, Fengmao Lv, Kai Lv, Olga Lyashevskaya, Chris Lydia, Maria Lymperaiou, Bohan Lyu, Chenyang Lyu, Chunchuan Lyu, Hanjia Lyu, Michael Lyu, Weimin Lyu, Xinglin Lyu, Xinxi Lyu, Yiwei Lyu, Yougang Lyu, Alexander Lyzhov

Bolei Ma, Chen Ma, Chenhao Ma, Fukun Ma, Guangyuan Ma, Haokai Ma, Ji Ma, Jing Ma, Longxuan Ma, Minghua Ma, Nianzu Ma, Pingchuan Ma, Qianli Ma, Qiyao Ma, Rao Ma, Shuo Ma, Weicheng Ma, Weitao Ma, Weizhi Ma, Xiaomeng Ma, Xinbei Ma, Xinge Ma, Xinyin Ma, Xueguang Ma, Youmi Ma, Yunpu Ma, Yuqing Ma, Zexiong Ma, Zhengrui Ma, Zhiyuan Ma, Zihan Ma, Ziyang Ma, Sashank Macha, Jakub Macina, Mounica Maddela, Marco Madeddu, Rahul Madhavan, Manas Madine, Andreas Madsen, Tharindu Madusanka, Joao Magalhaes, Walid Mag-

dy, Pierre Magistry, Bernardo Magnini, Khyati Mahajan, Shweti Mahajan, Kishan Maharaj, Sainik Kumar Mahata, Ridwan Mahbub, A V S D S Mahesh, Ayush Maheshwari, Gaurav Maheshwari, Rishabh Maheshwary, Aishwarya Maheswaran, Syed Hasan Amin Mahmood, Mohammad Mahoor, Sijie Mai, Arindam Majee, Navonil Majumder, Márton Makrai, Emanuele La Malfa, Manuj Malik, Advaith Malladi, Christopher Malon, Mamta Mamta, Zhibo Man, Sunny Manchanda, Avshalom Manevich, Varun Manjunatha, Christopher D. Manning, Ioana Manolescu, Jonibek Mansurov, Hangyu Mao, Kelong Mao, Qianren Mao, Rui Mao, Xiaoxi Mao, Yu Mao, Zhiming Mao, Zhuoyuan Mao, Guillermo Marco, Marion Di Marco, Alda Mari, Vukosi N. Marivate, Katja Markert, Ilia Markov, Shaul Markovitch, Elan Sopher Markowitz, Mounika Marreddy, Jason Mars, Federico Martelli, Anna Martin, James Martin, Lara J. Martin, Paul Martin, Matej Martinc, Richard Diehl Martinez, Eugenio Martinez-camara, Bruno Martins, Antonio David Ponce Martínez, Reem Masoud, Shahed Masoudian, Ahmed Masry, Sarah Masud, Ryo Masumura, Ved Mathai, Sandeep Mathias, Leena Mathur, Puneet Mathur, Yuji Matsumoto, Florian Matthes, Yevgen Matusevych, Evgeny Matusov, Kaushal Kumar Maurya, Costas Mavromatis, Diana Maynard, Philipp Mayr, Alessandro Mazzei, Sabrina Mccallum, R. Thomas Mccoy, John Philip Mccrae, Daniel Mcduff, Euan Mcgill, Nick Mckenna, Kathleen Mckeown, Nicholas Meade, Sourav Medya, Jannat Ara Meem, Michiel Van Der Meer, Alexander Mehler, Houman Mehrafarin, Kartik Mehta, Maitrey Mehta, Sanket Vaibhav Mehta, Jingbiao Mei, Lingrui Mei, Sen Mei, Stephen Meisenbacher, Abdellah El Mekki, Jesus Lovón Melgarejo, Elena Alvarez Mellado, Gerard Melo, Velayuthan Menan, Julia Mendelsohn, Ethan Adrian Mendes, Marcelo Mendoza, Arul Menezes, João Menezes, Chuan Meng, Lingyuan Meng, Xuying Meng, Yuanliang Meng, Stefano Menini, Rakesh R Menon, Wolfgang Menzel, Fabio Mercorio, Paola Merlo, Mike A Merrill, William Merrill, Rin Metcalf, Guillaume Metzler, Francois Meyer, Selina Meyer, Stefano Mezza, Md Messal Monem Miah, Yibo Miao, Yisong Miao, Alessio Miaschi, Kurt Micallef, James Michaelov, Zulfat Miftahutdinov, Rada Mihalcea, Tsvetomila Mihaylova, Vladislav Mikhailov, Jeremiah Milbauer, Filip Miletic, Aleksandra Miletić, Timothy A. Miller, David Mimno, Dehai Min, Do June Min, Qingkai Min, Mario Mina, Pasquale Minervini, Koji Mineshima, Tianshi Ming, Yifei Ming, Benjamin Minixhofer, Santiago Miret, Seyedabolghasem Mirroshandel, Aziza Mirsaidova, Mehrnoosh Mirtaheri, Abhijit Mishra, Aditya M. Mishra, Kshitij Mishra, Pruthwik Mishra, Amita Misra, Masato Mita, Cassie S. Mitchell, Avijit Mitra, Dr. Satanik Mitra, Ashish Mittal, Happy Mittal, Makoto Miwa, Taiki Miyanishi, Yusuke Miyao, Taro Miyazaki, Wenjie Mo, Ali Modarressi, Sandip Modha, Arkadiusz Modzelewski, Lucas Moeller, Abdelrahman Mohamed, Amr Mohamed, Youssef Sherif Mansour Mohamed, Manas Ranjan Mohanty, Jisoo Mok, Laxmikant Kishor Mokadam, Francesco Maria Molfese, Nicholas Monath, Erfan Moosavi Monazzah, Ishani Mondal, Mainack Mondal, Philipp Mondorf, Helena Silva Moniz, Adrian Pastor Lopez Monroy, Syed Mostofa Monsur, Stefano Montanelli, Arturo Montejo-ráez, Kyle Montgomery, Hyeonseok Moon, Suhong Moon, Ray Mooney, Guy Mor, Raha Moraffah, Gianmarco De Francisci Morales, Tetsuro Morimura, Luca Moroni, Jacob Daniel Morrison, Isaac Thani Moses, Viktor Moskvoretskii, Pablo Mosteiro, Josiane Mothe, Yutao Mou, Ali Mousavi, Seyed Mahed Mousavi, Emily Mower, Li Moxin, Maximilian Mozes, Honglin Mu, Yida Mu, Yongyu Mu, Hamdy Mubarak, Shamsuddeen Hassan Muhammad, Rohit Mujumdar, Animesh Mukherjee, Anjishnu Mukherjee, Sagnik Mukherjee, Jonathan Mukiibi, Matthew Mulholland, Carlos Mullov, Hyeongdon Mun, Rajesh Kumar Mundotiya, Vishvak Saivenkat Murahari, Koji Murakami, Soichiro Murakami, Masayasu Muraoka, Yugo Murawaki, Andrei Ioan Muresanu, Keerthiram Murugesan, Pradeep Kumar Murukannaiah, John Murzaku, Elio Musacchio, Tomáš Musil, Skatje Myers, Agnieszka Mykowiecka

Byeonghu Na, Cheolwon Na, Hongbin Na, Seung-hoon Na, Nona Naderi, Mohamed Nadif, Hubert Naets, Arijit Nag, Sayan Nag, Ryo Nagata, Md Mahadi Hasan Nahid, Aakanksha Naik, Atharva Naik, Inderjeet Jayakumar Nair, Rahul Nair, Tetsuji Nakagawa, Itay Nakash, Yuta Nakashima, Ndapa Nakashole, Makoto Nakatsuji, Eric Nalisnick, Akshay Nambi, Guoshun Nan, Federico Nanni, Tarek Naous, Diane Napolitano, Sharan Narasimhan, Shashi Narayan, Daniele Nardi, Ne-

ha Narwal, Jamal Nasir, Anandhavelu Natarajan, Abhijnan Nath, Shah Nawaz, Shravan Nayak, Tapas Nayak, Hamada Nayel, Renato Negrinho, Anil Kumar Nelakanti, Sree Harsha Nelaturu, Poli Nemkova, Vera Neplenbroek, Graham Neubig, Günter Neumann, Mariana Neves, Benjamin Newman, Jun-ping Ng, Raymond T. Ng, See-kiong Ng, Youyang Ng, Henry Nghiem, Nghia Trung Ngo, Anh Nguyen, Bang Nguyen, Chien Van Nguyen, Duc-vu Nguyen, Duy Nguyen, Giang Nguyen, Huy Nguyen, Khanh Xuan Nguyen, Kiem-hieu Nguyen, Long Hb Nguyen, Luan Thanh Nguyen, Minh Van Nguyen, Ngan Luu-thuy Nguyen, Thao Nguyen, Thong Nguyen, Tin Nguyen, Tin Trung Nguyen, Truc-vien T. Nguyen, Tuc Van Nguyen, Tung Nguyen, Van Bach Nguyen, Vincent Nguyen, Jingwei Ni, Junrui Ni, Xuanfan Ni, Yuansheng Ni, Zhaoheng Ni, Dong Nie, Ercong Nie, Jian-yun Nie, Pengyu Nie, Ping Nie, Yuxiang Nie, Zhijie Nie, Elizabeth Kaye Nielsen, Malvina Nikandrou, Anna Nikiforovskaya, Irina Nikishina, Christina Niklaus, Sergey I. Nikolenko, Jinzhong Ning, Kun-peng Ning, Ruoxi Ning, Noriki Nishida, Sergiu Nisioi, Malvina Nissim, Vikram Nitin, Chaoyue Niu, Guanglin Niu, Hao Niu, Jianwei Niu, Jingcheng Niu, Luyao Niu, Tianhao Niu, Tianyi Niu, Xueyan Niu, Joakim Nivre, Ayana Niwa, Bill Noble, Rodrigo Frassetto Nogueira, Keonwoong Noh, Adam Nohejl, Tadashi Nomoto, Rik Van Noord, Kai North, Marzia Nouri, Damien Nouvel, Scott Novotney, Sarana Nutanong, Ifeoma Nwogu

Alexander O'connor, Nicolás Benjamín Ocampo, Daniela Occhipinti, Brendan Oconnor, Yusuke Oda, Stephan Oepen, Owen Oertell, Alex Oesterling, Bahadorreza Ofoghi, Perez Atieno Ogayo, Michael Ogezi, Byung-doh Oh, Hanseok Oh, Hongseok Oh, Jaehoon Oh, Jungseul Ok, Tatsushi Oka, Yui Oka, Shu Okabe, Tsuyoshi Okita, Eda Okur, Tobi Olatunji, Hugo Gonçalo Oliveira, Adjali Omar, Ethel Chua Joy Ong, Subba Reddy Oota, Andreas Opedal, Juri Opitz, Alina Oprea, Maite Oronoz, John Ortega, Naoki Otani, Yulia Otmakhova, Jiao Ou, Jiefu Ou, Yixin Ou, Zhijian Ou, Siqi Ouyang, Siru Ouyang, Yang Ouyang, Zhongyu Ouyang, Risako Owan, Momose Oyama, Masafumi Oyamada, Savas Ozkan, Arzucan Özgür

Inkit Padhi, Trilok Padhi, Tejomay Kishor Padole, Aline Paes, Artidoro Pagnoni, Vardaan Pahuja, Partha Pakray, Christopher J Pal, Kuntal Kumar Pal, Proyag Pal, Chester Palen-michel, Matteo Palmonari, Flavio Di Palo, John Palowitch, Shramay Palta, Bo Pan, Deng Pan, Huitong Pan, Jane Pan, Junting Pan, Rong Pan, Rui Pan, Ruihao Pan, Shirui Pan, Xiang Pan, Xiao Pan, Yanzhou Pan, Youcheng Pan, Zhenyu Pan, Artemis Panagopoulou, Ruchi Panchanadikar, Alexander Panchenko, Pranoy Panda, Subhadarshi Panda, Nilay Pande, Punya Syon Pandey, Saurabh Kumar Pandey, Shrey Pandit, Chenxi Pang, Tianyu Pang, Sandeep Kumar Pani, Abhishek Panigrahi, Panlei, Aleksandr Panov, Evangelos E. Papalexakis, Sean Papay, Paolo Papotti, Duccio Pappadopulo, Ashraful Islam Paran, Natalie Parde, Amit Parekh, Tanmay Parekh, Angelina Parfenova, Priyank Parikh, Chaehun Park, Chan Young Park, Cheonbok Park, Cheoneum Park, Choonghyun Park, Eunhwan Park, Eunhyeok Park, Gyeong-moon Park, Hogun Park, Jinyoung Park, Jun-hyung Park, Jungsoo Park, Jungyeul Park, Seongmin Park, Shinwoo Park, Sungwon Park, Mihir Parmar, Marinela Parovic, Pranjali Ajay Parse, Md Rizwan Parvez, Ankita Pasad, Nivranshu Pasricha, Valeria Pastorino, Ajay Patel, Hitesh Laxmichand Patel, Maitreya Patel, Nilay Patel, Sarang Rajendra Patil, Vaidehi Patil, Jasabanta Patro, Priyaranjan Pattnayak, Parth Patwa, Manasi Patwardhan, Indraneil Arun Paul, Mithun Paul, Shounak Paul, Amalie Brogaard Pauli, John Pavlopoulos, Vera Pavlova, Sachin Pawar, Siddhesh Pawar, Ali Payani, Patomporn Payoungkhamdee, Tzuf Paz-argaman, Branislav Pecher, Pavel Pecina, Changhua Pei, Qizhi Pei, Georgios Peikos, Catherine Pelachaud, Lotem Peled-cohen, Kellin Pelrine, Abhirama Subramanyam Penamakuri, Baolin Peng, Bo Peng, Dezhi Peng, Dezhong Peng, Jiawei Peng, Jing Peng, Keqin Peng, Letian Peng, Lim Jia Peng, Miao Peng, Min Peng, Qiming Peng, Qiwei Peng, Qiyao Peng, Shaohui Peng, Tianhao Peng, Tianshuo Peng, Wei Peng, Wen-chih Peng, Xi Peng, Xiangyu Peng, Yifan Peng, Yifan Peng, Gerald Penn, Valerio Pepe, Joseph Joshua Peper, Gal Peretz, Jesus Javier Calleja Perez, Laura Perez-beltrachini, Gabriele Pergola, Nathanaël Perraudin, Stefano Perrella, Ali Pesaranghader, Stanislav Peshterliev, Denis Peskov, Jan-thorsten Peter, Ben Peters, Pavel Petrushkov, Pouya Pezeshkpour, Hanspeter Pfister,

Quang Hieu Pham, Fred Philippy, Renjie Pi, Yinhua Piao, Massimo Piccardi, Janet B. Pierrehumbert, Michał Pietruszka, Reshmi Gopalakrishna Pillai, Mathis Pink, Ramakrishna Pinninti, Bartosz Piotrowski, Andrew Piper, Hugo Pitorro, Priya Pitre, Lidia Pivovarova, Paul Piwek, Barbara Plank, Moritz Plenz, Max Ploner, Soham Poddar, Christian Poelitz, Thierry Poibeau, Rhitabrat Pokharel, Sandeep Polisetty, Ramesh Poluru, Josiah Poon, Adrian Popescu, Nicholas Popovic, Ian Porada, Beatrice Portelli, François Portet, Matt Post, Martin Potthast, Amrit Poudel, Mohammadreza Pourreza, Manuel Prada-corral, Ritu Prajapati, Aniket Pramanick, A Pranav, Jakob Prange, Shivika Prasanna, Anurag Pratik, Gabriele Prato, Ian Pratt-hartmann, Frederic Precioso, Damith Premasiri, Valentina Presutti, Sarah Masud Preum, Priyanshu Priya, Lorenzo Proietti, Victor Prokhorov, Ruili Pu, Shi Pu, Yewen Pu, Yifan Pu, Giovanni Puccetti, Giulia Pucci, Ratish Puduppully, Saurabh Pujar, Rajkumar Pujari, Sukannya Purkayastha, Alberto Purpura, Matthew Purver, Ayu Purwarianti, Rifki Afina Putri, Meghana Puvvadi, Juan Antonio Pérez-ortiz

Ayesha Qamar, Jianzhong Qi, Jirui Qi, Peng Qi, Weijian Qi, Xuan Qi, Yunjia Qi, Dong Qian, Hongjin Qian, Kun Qian, Xueming Qian, Yining Qian, Yusu Qian, Zhaozhi Qian, Zhenxing Qian, Zhong Qian, Jipeng Qiang, Minjie Qiang, Zewen Qiang, Shanbao Qiao, Shuofei Qiao, Bowen Qin, Chengwei Qin, Haotong Qin, Jianbin Qin, Jinghui Qin, Tian Qin, Wei Qin, Yuehan Qin, Zhanyue Qin, Zhenyue Qin, Peijun Qing, Huachuan Qiu, Lin Qiu, Linlu Qiu, Mengyang Qiu, Xihe Qiu, Xin Ying Qiu, Yifu Qiu, Zhaopeng Qiu, Zihan Qiu, Muhammad Reza Qorib, Jiaxing Qu, Xiaoye Qu, Xingwei Qu, Yanru Qu, Yincen Qu, Yuzhong Qu, Xin Quan, Ariadna Quattoni, Haoran Que, Solen Quiniou, Mohammed Rameez Qureshi, Rizwan Qureshi

Gowtham R, Narendhiran R, Reihaneh Rabbany, Hamid R. Rabiee, Ella Rabinovich, Karthik Radhakrishnan, Bhaktipriya Radharapu, Daniele P. Radicioni, Taki Hasan Rafi, Davood Rafiei, Alessandro Raganato, Luca Ragazzi, Vipul Raheja, Vahid Rahimzadeh, Mizanur Rahman, Shafin Rahman, Hossein A. Rahmani, Nishat Raihan, Tina Raissi, Lisa Raithel, Chahat Raj, Sahar Rajabi, Sara Rajaee, Kanagasabai Rajaraman, Rishanth Rajendhran, Shahab Raji, Sarah Michele Rajtmajer, Nathanaël Carraz Rakotonirina, Anil Ramakrishna, Bhaskar Ramasubramanian, Giulia Rambelli, Owen Rambow, Aida Ramezani, Sahana Ramnath, Sanjana Ramprasad, Shivaen Ramshetty, Pritika Ramu, Guillem Ramírez, Shihao Ran, Yide Ran, Ashish Rana, Federico Ranaldi, Anku Rani, Dongning Rao, Gengyu Rao, Jun Rao, Rajath Rao, Zhi Rao, Ahmad Rashid, Vipul Kumar Rathore, Shivam Raval, Federico Ravenda, Sahithya Ravi, Ajay Madhavan Ravichandran, Renjith P Ravindran, Yogesh S Rawat, Sourjyadip Ray, Simon Razniewski, Olesya Razuvayevskaya, Imran Razzak, Varshini Reddy, Chris Reed, Michaela Regneri, Abudurexiti Reheman, Nils Reiter, Sravan Reddy Rekula, Houxing Ren, Jie Ren, Jie Ren, Kan Ren, Kui Ren, Qingyang Ren, Runtao Ren, Shuo Ren, Weijieying Ren, Xuanfei Ren, Xubin Ren, Yubing Ren, Yujie Ren, Yarik Menchaca Resendiz, Achim Rettinger, Anja Reusch, Artem Revenko, Mohammadhossein Rezaei, Rezvaneh Rezapour, Kathryn Ricci, Kyle Richardson, Stephen D. Richardson, Korbinian Riedhammer, Maarten De Rijke, Laura Rimell, Md Imbesat Hassan Rizvi, Jesse Roberts, Gabriel Roccabruna, Gil Rocha, Julian Martin Rodemann, Christophe Rodrigues, Pedro Rodriguez, Melissa Roemmele, Lina Maria Rojas-barahona, Nathan Roll, Wenge Rong, Jonathan Rose, Mohammad Rostami, Aiala Rosá, Guy Rotman, Kay Rottmann, Dmitri Roussinov, Bryan R. Routledge, Amartya Roy, Amit Roy, Aniruddha Roy, Shamik Roy, Soumyadeep Roy, Sudipta Singha Roy, Nikolai Rozanov, Chong Ruan, Jiacheng Ruan, Jie Ruan, Jingqing Ruan, Tong Ruan, Terry Ruas, Patrick Ruch, William Rudman, Daniel Ruffinelli, Federico Ruggeri, Ruibochen, Mukund Rungta, Hu Ruofan, Jonathan Rusert, Stefan Ruseti, Jenna Russell, Daniel Russo, Irene Russo, Derek Ruths, Elena Sofia Ruzzetti, Pavel Rychlý, Sangwon Ryu, Rafal Rzepka, Susanna Rücker

Ramaneswaran S, Jon Saad-falcon, Ashish Sabharwal, Ahmed Sabir, Sahand Sabour, Rachneet Singh Sachdeva, Kaushik Ram Sadagopan, Abdelrahman Boda Sadallah, Mobashir Sadat, Amir Saeidi, Takaaki Saeki, Bardia Safaei, Mustafa Safdari, Benoît Sagot, Amit Kumar Sah, Avirup

Saha, Punyajoy Saha, Rohan Saha, Sougata Saha, Soumya Saha, Sriparna Saha, Tulika Saha, Sattvik Sahai, Vaibhav Sahai, Magnus Sahlgren, Nihar Ranjan Sahoo, Oscar Sainz, Yusuke Sakai, Andrey Sakhovskiy, Alireza Salemi, Jonne Saleva, Vishal Vivek Saley, Suchir Salhan, Flora Salim, Avneesh Saluja, Tanja Samardzic, Ranbir Singh Sanasam, Javier Iranzo Sanchez, Jivnesh Sandhan, Raghav Sandhu, Erik F. Tjong Kim Sang, Samin Mahdizadeh Sani, Maziar Sanjabi, Ashwin Sankar, Chinnadhurai Sankar, Aruna Sankaranarayanan, Scott Sanner, Leonid Sanochkin, Selene Baez Santamaria, Sashank Santhanam, Andrea Santilli, Bishal Santra, Debarshi Kumar Sanyal, Irina Saparina, Xabier Saralegi, Vasanth Sarathy, Gabriel Herbert Sarch, Arindam Sarkar, Rupak Sarkar, Souvika Sarkar, Sujoy Sarkar, Somdeb Sarkhel, Shota Sasaki, David Sasu, Ranjan Satapathy, Prasanna Sattigeri, Agata Savary, Andrey Savchenko, Beatrice Savoldi, Guergana K Savova, Manolis Savva, Apoorv Umang Saxena, Rohit Saxena, Michael Saxon, Salim Sazzed, Daniel Scalena, Carolina Scarton, Elizabeth J. Schaefer, Markus Schedl, Tatjana Scheffler, Marten Van Schijndel, David Schlangen, Helmut Schmid, Fabian David Schmidt, Patrícia Schmidtová, Vera Schmitt, Florian Schneider, Gerold Schneider, Julian Moreno Schneider, Nathan Schneider, Patrick Schramowski, Christopher Schröder, Dan Schumacher, Elliot Schumacher, Jakob Schuster, Stefan Schweter, Ulrich Schäfer, Simeon Schüz, Alessandro Scirè, Amit Seker, Bilgehan Sel, David Semedo, Sina Semnani, Jaydeep Sen, Meghdut Sengupta, Jaehyung Seo, Jeongyeon Seo, Sofia Serrano, Sergio Servantez, Agrima Seth, Agam Shah, Alay Dilipbhai Shah, Mubarak Shah, Shalin Shah, Syed Afaq Ali Shah, Haz Sameen Shahgir, Rifat Shahriyar, Rana Shahroz, Chantal Shaib, Sagi Shaier, Omar Shaikh, Cory Shain, Gregory Shakhnarovich, Valerie Shalin, Hengchao Shang, Lanyu Shang, Ziyao Shangguan, Chen Shani, Abhilash Shankarampeta, Chenze Shao, Jing Shao, Wei Shao, Wenqi Shao, Yanjun Shao, Yihua Shao, Yijia Shao, Yingxia Shao, Yunfan Shao, Nada Sharaf, Omar Sharif, Abhishek Sharma, Divya V Sharma, Kartik Sharma, Mohit Sharma, Prashant Kumar Sharma, Raksha Sharma, Roshan Sharma, Behzad Shayegh, Artem Shelmanov, Bowen Shen, Huanming Shen, Huawei Shen, Jiaxin Shen, Jiayi Shen, Jocelyn J Shen, Li Shen, Linlin Shen, Ming Shen, Siqi Shen, Tao Shen, Tianhao Shen, Xiangqing Shen, Xingyu Shen, Xinyue Shen, Yatian Shen, Ying Shen, Ying Shen, Zejiang Shen, Zhan Shen, Emily Sheng, Jiawei Sheng, Qiang Sheng, Quan Z. Sheng, Zhang Jia Sheng, Shengbinyue, Ivaxi Sheth, Anudeex Shetty, Pooja Shetty, Prakriti Shankar Shetty, Chufan Shi, Dachuan Shi, Daqian Shi, Haochen Shi, Haoxiang Shi, Jiatong Shi, Jiatong Shi, Jie Shi, Kejian Shi, Lida Shi, Qi Shi, Taiwei Shi, Tianyuan Shi, Weijia Shi, Wenqi Shi, Xiao Shi, Xiaoming Shi, Xiaoshuang Shi, Yaorui Shi, Yaya Shi, Yucheng Shi, Zesheng Shi, Zhan Shi, Zhengxiang Shi, Zijing Shi, Tomohide Shibata, Min-han Shih, Li Shilong, Hidetoshi Shimodaira, Erica Kido Shimomoto, Anastasia Shimorina, Haebin Shin, Hyopil Shin, Hyunseo Shin, Jisu Shin, Seungwon Shin, Kazutoshi Shinoda, Kiyoaki Shirai, Yow-ting Shiue, Pannaga Shivaswamy, Michal M Shlapentokh-rothman, Vitaly Shmatikov, Michal Shmueli-scheuer, Lidan Shou, Kumar Shridhar, Kai Shu, Tianmin Shu, Yao Shu, Yiheng Shu, Hong-han Shuai, Wang Shuai, Kai Shuang, Liu Shudong, Ruihao Shui, Zeren Shui, Shreya Shukla, Utsav Shukla, Kashun Shum, Ammon C Shurtz, Shuwen, Chenglei Si, Shijing Si, Shuzheng Si, Anthony Sicilia, Lucia Siciliani, A.b. Siddique, Grigori Sidorov, Noah Yamamoto Siegel, Zhou Sijia, Sandipan Sikdar, Damien Sileo, Eliezer De Souza Da Silva, Nisansa De Silva, Purificação Silvano, Fabrizio Silvestri, Patrick Simianer, Antoine Simoulin, Elena Simperl, Amrita Singh, Apoorva Singh, Avyav Kumar Singh, Ayushman Singh, Bhavyajeet Singh, Chandan Singh, Janvijay Singh, Joykirat Singh, Mantek Singh, Mukul Singh, Pranaydeep Singh, Prateek Singh, Rituraj Singh, Sonali Singh, Mohit Singhal, Iustin Sirbu, Sonish Sivarajkumar, Johan Sjons, Steven Skiena, Gabriella Skitalinskaya, Paweł Marek Skórzewski, Charese Smiley, David A. Smith, Răzvan-alexandru Smădu, Vésteinn Snæbjarnarson, Daria Soboleva, Jy-yong Sohn, Felix Soldner, Taiga Someya, Pia Sommerauer, Bokyung Son, Junyoung Son, Seungwoo Son, Chan Hee Song, Chunyao Song, Dandan Song, Dawei Song, Dawn Song, Feifan Song, Guojie Song, Haiyue Song, Hoyun Song, Hwanjun Song, Hyun-je Song, Jongyoon Song, Kaisong Song, Kaitao Song, Kehui Song, Kyungwoo Song, Lingran Song, Lingyun Song, Linqi Song, Linxin Song, Mingyang Song, Mingyang Song, Peiyang Song, Ran Song, Rui Song, Ruihua Song, Xiaoshuai Song, Xin Song,

Yingjin Song, Yiping Song, Yuanfeng Song, Yueqi Song, Yun-zhu Song, Yurun Song, Zhao Song, Zhenxi Song, Zifan Song, Zirui Song, Atin Sood, Lay-ki Soon, Alessandro Sordoni, Jeffrey S. Sorensen, Taylor Sorensen, Ionut Sorodoc, Alexey Sorokin, Tiberiu Sosea, Anna Sotnikova, Alvaro Soto, Rafael Alberto Rivera Soto, Sajad Sotudeh, Zhivar Sourati, Dilara Soylu, Alexander Spangher, Lucia Specia, Alessandro Sperduti, Michal Spiegel, Andreas Spitz, Maximilian Spliethöver, Zayne Rea Sprague, Ivan Srba, Makesh Narsimhan Sreedhar, Neha Srikanth, Mukund Srinath, Dananjay Srinivas, Jayanth Srinivasa, Tejas Srinivasan, Vijay Srinivasan, Ankit Kumar Srivastava, Aseem Srivastava, Dhruv Srivastava, Mani Srivastava, Saurabh Srivastava, Kv Aditya Srivatsa, Joe Stacey, Ieva Staliunaite, Dominik Stammbach, Giorgos B. Stamou, Karolina Stanczak, Katherine Stasaski, Mark Steedman, Julius Steen, Nicolas Stefanovitch, Sebastian Steindl, Andreas Stephan, Igor Sterner, Hannah Sterz, Ian Stewart, Asa Cooper Stickland, Shane Storks, Svetlana Stoyanchev, Ivelina Stoyanova, Sebastian Stüker, Guixin Su, Hang Su, Hao Su, Hongjin Su, Hung-ting Su, Puzhen Su, Sen Su, Weihang Su, Xiangdong Su, Xin Su, Yi Su, Yongye Su, Zhaochen Su, Zhe Su, Zhenpeng Su, D. Subhalingam, Malliga Subramanian, Prasanna Lakkur Subramanyam, Peiqi Sui, Xuhui Sui, Yuan Sui, Bin Sun, Changzhi Sun, Chengjie Sun, Chenkai Sun, Dachun Sun, Guangzhi Sun, Guangzhong Sun, Hanshi Sun, Hao Sun, Haotian Sun, Huaman Sun, Jingchen Sun, Jingyi Sun, Jun Sun, Kai Sun, Kaiser Sun, Ke Sun, Kexuan Sun, Kun Sun, Ling Sun, Linzhuang Sun, Manxi Sun, Mengshu Sun, Penglei Sun, Qiao Sun, Qiushi Sun, Renliang Sun, Rui Sun, Ruoyu Sun, Shiliang Sun, Si Sun, Siqi Sun, Tiening Sun, Wei Sun, Xiaobing Sun, Xiaoshuai Sun, Yang Sun, Yuan Sun, Yuanyuan Sun, Yuchong Sun, Yuqing Sun, Yuxi Sun, Zechen Sun, Zengkui Sun, Zequn Sun, Zexu Sun, Zhaoyue Sun, Zhewei Sun, Zhongxiang Sun, Zhouhao Sun, Anirudh S Sundar, Jayaprakash Sundararaj, Megha Sundriyal, Hakyung Sung, Mujeen Sung, Yoo Yeon Sung, Sunhao, Monica Sunkara, Marek Suppa, Siddharth Suresh, Tarun Suresh, Manan Suri, Aditya Kaushik Surikuchi, Simon Suster, Lintang Sutawika, Yoshimi Suzuki, Pedro Javier Ortiz Suárez, Anej Svete, Colin Swaelens, Sarathkrishna Swaminathan, Swabha Swayamdipta, Zsolt Szanto, Andreas Säuberli, Mihailo Škorić

Jaesung Tae, Shabnam Tafreshi, Andrea Tagarelli, Marzieh S. Tahaei, Francesco Taioli, Yu Takagi, Takehiro Takayanagi, Koichi Takeuchi, Tetsuya Takiguchi, Ece Takmaz, Ekaterina Taktasheva, Zhi Rui Tam, Ales Tamchyna, Chao-hong Tan, Cheng Tan, Cheston Tan, Chuanyuan Tan, Fei Tan, Hanzhuo Tan, Haochen Tan, Hongye Tan, Qiaoyu Tan, Shaomu Tan, Shiwei Tan, Weiting Tan, Xiaoyu Tan, Xin Tan, Yusong Tan, Zeqi Tan, Zhen Tan, Zhentao Tan, Shohei Tanaka, Bo Tang, Buzhou Tang, Chen Tang, Chenming Tang, Haoran Tang, Jing Tang, Jing Tang, Kevin Tang, Kewen Tang, Liyan Tang, Lumingyuan Tang, Nan Tang, Ningzhi Tang, Raphael Tang, Ruixiang Tang, Wei Tang, Xiaohang Tang, Xuemei Tang, Xunzhu Tang, Yihong Tang, Yixuan Tang, Yixuan Tang, Yu-chien Tang, Yubao Tang, Yun Tang, Zecheng Tang, Zheng Tang, Zhenheng Tang, Zhiwen Tang, Zichen Tang, Zilu Tang, Kumar Tanmay, Chris Tanner, Farhan Tanvir, Eshaan Tanwar, Chaofan Tao, Chongyang Tao, Guanhong Tao, Jianhua Tao, Mingxu Tao, Wenbiao Tao, Xijia Tao, Zhang Wen Tao, Zhengwei Tao, Zhiqiang Tao, Zhuo Tao, Maksym Taranukhin, Stephen Eugene Taylor, Diane Tchuindjo, Matthieu Tehenan, Selma Tekir, Atharva Tendle, Damien Teney, Yan Teng, Simone Teufel, Katherine Thai, Patricia Thaine, Megh Thakkar, Surendrabikram Thapa, Ashish V Thapliyal, Ramya Keerthy Thatikonda, Anirudh Thatipelli, Barry-john Theobald, Sarubi Thillainathan, Craig Alexander Thomson, James Thorne, Veronika Thost, Poojitha Thota, David Thulke, Bozhong Tian, Chang Tian, Jiahao Tian, Jidong Tian, Jinchuan Tian, Junfeng Tian, Qi Tian, Shiyu Tian, Shulin Tian, Yapeng Tian, Yu Tian, Yuan Tian, Yuchen Tian, Roelien C Timmer, Abhisek Tiwari, Lindia Tjuatja, Antonela Tommasel, Hanghang Tong, Hao Tong, Meihan Tong, Xiaoyu Tong, Jonathan Tonglet, Atnafu Lambebo Tonja, Manuel Tonneau, Marwan Torki, Rishit Toteja, Samia Touileb, Julien Tourille, Itamar Trainin, Hieu Tran, Khiem Vinh Tran, Khoi-nguyen Tran, Quyen Tran, Son Quoc Tran, Marco Del Tredici, Volker Tresp, Ilias Triantafyllopoulos, Jan Trienes, Remi Van Trijp, Rocco Tripodi, Ashutosh Trivedi, Harsh Trivedi, Shubhendu Trivedi, Yun Da Tsai, Yu Tsao, Dimitrios Tsarapatsanis, Bo-hsiang Tseng, Shao-yen Tseng, Talia Tseriotou,

Ioannis Tsiamas, Eleftheria Tsipidi, Hayato Tsukagoshi, Akim Tsvigun, Lifu Tu, Mei Tu, Quan Tu, Rong-cheng Tu, Shangqing Tu, Elena Tutubalina, Aman Tyagi, Utkarsh Tyagi

Takuma Udagawa, Akhil Udathu, Can Udomcharoenchaikit, Ryo Ueda, Gökçe Uludoğan, Alexandra Uma, Inigo Jauregi Unanue, Rheeya Uppaal, Ashok Urlana, Nicolas Usunier, Benno Uthayasooriyar, Takehito Utsuro, Ozlem Uzuner

Robert Vacareanu, Saujas Vaduguru, Teemu Vahtola, Ashwini Vaidya, Sowmya Vajjala, Nidhi Vakil, Francisco Valentini, Marco Valentino, Andric Valdez Valenzuela, Estrella Vallecillo-rodríguez, Sai P Vallurupalli, Hoang Van, Linh Ngo Van, Vincent Vandeghinste, Natalia Vanetik, Vasudha Varadarajan, Siddharth Varia, Shiva Maleki Varnosfaderani, Deeksha Varshney, Prasoon Varshney, Rajashekar Vasantha, Aniket Vashishtha, Oleg Vasilyev, Justin Vasselli, Pavlo Vasylenko, Artem Vazhentsev, Eva Maria Vecchi, Nikhita Vedula, Hariram Veeramani, Dan John Velasco, Erik Velldal, Raviteja Vemulapalli, Suresh Venkatasubramanian, Ganesh Venkatesh, Kavana Venkatesh, Praveen Venkateswaran, Apurv Verma, Arun Verma, Neha Verma, Rajeev Verma, Yash Verma, Keith Vertanen, Gianluca Vico, Renata Vieira, Lovekesh Vig, Anvesh Rao Vijjini, David Vilares, Dan Vilenchik, Danae Sanchez Villegas, Vibhav Vineet, Krishnapriya Vishnubhotla, Amar Viswanathan, Vijay Viswanathan, Aditya Visweswaran, Nikolas Vitsakis, Jean-noël Vittaut, Andreas Vlachos, Kiran Vodrahalli, Ekaterina Voloshina, Pavlos Vougiouklis, Duy-khanh Vu, Huy-the Vu, Trang Vu, Renato Vukovic, Ugurcan Vurgun, Ivan Vykopal, Ekaterina Vylomova, Horacio Jarquín Vásquez, Juan Vásquez, Raúl Vázquez

Takashi Wada, Sahil Wadhwa, Somin Wadhwa, Claudia Wagner, Eitan Wagner, Joachim Wagner, Abdul Waheed, Jan Philip Wahle, Eric Peter Wairagala, Hiromi Wakaki, Sabine Schulte Im Walde, Andreas Waldis, Jonas Wallat, Changxuan Wan, Chenwei Wan, Fanqi Wan, Hai Wan, Herun Wan, Huaiyu Wan, Kaiyang Wan, Qizhi Wan, Stephen Wan, Yao Wan, Yingjia Alisa Wan, Yixin Wan, Zhen Wan, Zhongwei Wan, An Wang, Bang Wang, Baoxin Wang, Bichen Wang, Bing Wang, Bingbing Wang, Bingjie Wang, Bingqing Wang, Bo Wang, Bo Wang, Boxiang Wang, Chaojie Wang, Chaokun Wang, Chen Wang, Chen Wang, Cheng Wang, Chenglin Wang, Chenglong Wang, Chenxu Wang, Cong Wang, Cunxiang Wang, Daisy Zhe Wang, Daling Wang, Danqing Wang, Darren Yow-bang Wang, Di Wang, Di Wang, Dingmin Wang, Dingzirui Wang, Dongxia Wang, Fali Wang, Fan Wang, Gaoang Wang, Hai Wang, Haishuai Wang, Han Wang, Hao Wang, Hao Wang, Hao Wang, Haobo Wang, Haochun Wang, Haohan Wang, Haoran Wang, Haoran Wang, Haozhao Wang, Hebin Wang, Heng Wang, Heng Wang, Hengyi Wang, Hongning Wang, Hongru Wang, Hongya Wang, Houfeng Wang, Huadong Wang, Huazheng Wang, Huimin Wang, Huimu Wang, Ji Wang, Jia-ning Wang, Jialu Wang, Jian Wang, Jian Wang, Jiankun Wang, Jianyong Wang, Jianzong Wang, Jiapeng Wang, Jiaqi Wang, Jiashuo Wang, Jiawei Wang, Jiawei Wang, Jiayi Wang, Jie Wang, Jingwei Wang, Jingyu Wang, Jinqiao Wang, Jun Wang, Jun Wang, Junda Wang, Junlin Wang, Junsheng Wang, Junzhe Wang, Kai Wang, Kangrui Wang, Ke Wang, Kexiang Wang, Kexin Wang, Lanjun Wang, Lanrui Wang, Lei Wang, Li Wang, Liang Wang, Lianxi Wang, Lin Wang, Linlin Wang, Longbiao Wang, Lu Wang, Mengna Wang, Mengru Wang, Mengyu Wang, Ming Wang, Mingyang Wang, Pan Wang, Peiqi Wang, Peng Wang, Pengcheng Wang, Ping Wang, Qiang Wang, Qiang Wang, Qianli Wang, Qianlong Wang, Qianning Wang, Qing Wang, Qingni Wang, Qiqi Wang, Qiufeng Wang, Quan Wang, Renxi Wang, Rui Wang, Rui Wang, Ruibo Wang, Ruida Wang, Runhui Wang, Runze Wang, Ruoyao Wang, Shangfei Wang, Sheng Wang, Shengyuan Wang, Shida Wang, Shihan Wang, Shiqi Wang, Shuang Wang, Shuo Wang, Shuting Wang, Siqi Wang, Siqi Wang, Sirui Wang, Siyin Wang, Song Wang, Song Wang, Suge Wang, Suyuchen Wang, Tianchun Wang, Tianduo Wang, Tianshu Wang, Wei Wang, Wei-yao Wang, Weichuan Wang, Weixuan Wang, Weiyun Wang, Wen Wang, Wenjin Wang, Wenxuan Wang, Xi Wang, Xiang Wang, Xiangfeng Wang, Xianquan Wang, Xiao Wang, Xiaochen Wang, Xiaodong Wang, Xiaohan Wang, Xiaohua Wang, Xiaojie Wang, Xiaoling Wang, Xiaolong Wang, Xiaoqiang Wang, Xiawei

Wang, Xiaxia Wang, Xin Wang, Xin Wang, Xinfeng Wang, Xinglin Wang, Xinpeng Wang, Xintong Wang, Xinyi Wang, Xinyou Wang, Xinyu Wang, Xinyu Wang, Xinze Wang, Xiting Wang, Xu Wang, Xueqian Wang, Xunzhi Wang, Xuwu Wang, Yanbo Wang, Yanfeng Wang, Yang Wang, Yanhao Wang, Yanling Wang, Yaoming Wang, Yaqing Wang, Yasheng Wang, Ye Wang, Ye Wang, Ye Wang, Yibo Wang, Yichao Wang, Yichen Wang, Yifan Wang, Yifan Wang, Yihan Wang, Yijue Wang, Yike Wang, Yikun Wang, Yiming Wang, Yiming Wang, Yimu Wang, Yinggui Wang, Yining Wang, Yiran Wang, Yixu Wang, Yixuan Wang, Yixuan Wang, Yongjie Wang, Yu Wang, Yuchi Wang, Yue Wang, Yue Wang, Yue Wang, Yueqian Wang, Yuhang Wang, Yuhao Wang, Yuhao Wang, Yujie Wang, Yujie Wang, Yun Cheng Wang, Yuqi Wang, Yusong Wang, Yutong Wang, Yuxia Wang, Yuxin Wang, Yuxuan Wang, Zecheng Wang, Zekun Wang, Zekun Wang, Zekun Wang, Zezhong Wang, Zhaoyang Wang, Zhen Wang, Zheng Wang, Zheng Wang, Zhenglin Wang, Zhenhailong Wang, Zhenting Wang, Zhenyi Wang, Zhichun Wang, Zhilin Wang, Zhiruo Wang, Zhixiang Wang, Zhiyuan Wang, Zhizheng Wang, Zhizhi Wang, Zhongjie Wang, Zhuoer Wang, Zichao Wang, Zifu Wang, Zihan Wang, Zihan Wang, Zihan Wang, Zihao Wang, Zihao Wang, Ziheng Wang, Zijie Wang, Zili Wang, Zilong Wang, Zimu Wang, Ziqi Wang, Zirui Wang, Ziyue Wang, Wanli.li, Lilian Diana Awuor Wanzare, Nigel G. Ward, Greta Warren, Christian Wartena, Ishaan Watts, Ruvan Weerasinghe, Tharindu Cyril Weerasooriya, Anna Wegmann, Bowen Wei, Daimeng Wei, Fangyun Wei, Jerry Weng Wei, Jingxuan Wei, Johnny Wei, Kaiwen Wei, Kangda Wei, Lingwei Wei, Penghui Wei, Sheng-lun Wei, Xiao Wei, Xiaoyong Wei, Ying Wei, Yuancheng Wei, Yuchen Wei, Zhihua Wei, Zhongyu Wei, Zihao Wei, Gerhard Weikum, Maxwell Weinzierl, Fei Weizhi, Ji-rong Wen, Kaiyue Wen, Liang Wen, Xiaofei Wen, Xie Jia Wen, Ximing Wen, Yanlong Wen, Yule Wen, Yuqiao Wen, Zhiyuan Wen, Zhiyuan Wen, Wenbosi, Yixuan Weng, Peter West, Joyce Jiyoung Whang, Steven Euijong Whang, Haryo Akbarianto Wibowo, Philipp Wicke, Rachel Wicks, Michael Wiegand, Matti Wiegmann, Adam Wiemerslage, Ethan Wilcox, Bryan Wilie, Miles Williams, Steven R. Wilson, Guillaume Wisniewski, Sondre Wold, Tomer Wolfson, Ngai Wong, Wichayaporn Wongkamjan, Honguk Woo, Zach Wood-doughty, Phil Woodland, Anna Wróblewska, Alexander Wu, Anne Wu, Ben Peng Wu, Bin Wu, Bohao Wu, Bowen Wu, Changli Wu, Cheng-kuang Wu, Di Wu, Di Wu, Fang Wu, Feijie Wu, Han Wu, Haolun Wu, Haoyi Wu, Haoyuan Wu, Honghan Wu, Hongyan Wu, Hongyi Wu, Hui Wu, Ji Wu, Jiageng Wu, Jialong Wu, Jian Wu, Jiancan Wu, Jiawei Wu, Jiaying Wu, Jie Wu, Jincenzi Wu, Jinge Wu, Junchao Wu, Junhong Wu, Junjie Wu, Junkang Wu, Kaixin Wu, Lianwei Wu, Likang Wu, Linjuan Wu, Mengyue Wu, Minghao Wu, Mingyan Wu, Muling Wu, Peilin Wu, Peng Wu, Qianhui Wu, Qingqiang Wu, Qinzhuo Wu, Sai Wu, Shangyu Wu, Shengguang Wu, Shih-hung Wu, Shu Wu, Shujin Wu, Sifan Wu, Sixing Wu, Taiqiang Wu, Ting-wei Wu, Tongtong Wu, Tsung Han Wu, Wei Wu, Weiqi Wu, Weiyi Wu, Winston Wu, Xianchao Wu, Xin Wu, Xindong Wu, Xiyang Wu, Xuansheng Wu, Xueqing Wu, Xuyang Wu, Yang Wu, Yang Wu, Yangjun Wu, Yating Wu, Yawen Wu, Yifan Wu, Yihan Wu, Yimeng Wu, Yin Wu, Yingnian Wu, Yixin Wu, Yuan Wu, Yuanhao Wu, Yurong Wu, Yuwei Wu, Yuxia Wu, Zekun Wu, Zengqing Wu, Zhaoxuan Wu, Zhijing Wu, Zhonghai Wu, Zhuoyang Wu, Zihui Wu, Zijun Wu, Zirui Wu, Zongru Wu, Zongyu Wu, Jörn Wübker

Wang Xi, Zhaohan Xi, Haotian Xia, Jingbo Xia, Lei Xia, Lianghao Xia, Peng Xia, Qingrong Xia, Rui Xia, Tian Xia, Tingyu Xia, Wu Xia, Xiao Xia, Yan Xia, Yu Xia, Hu Xiang, Jiannan Xiang, Lu Xiang, Wei Xiang, Yue Xianghu, Xi Xiangyu, Bei Xiao, Chaojun Xiao, Chenghao Xiao, Chuan Xiao, Chunyang Xiao, Jin Xiao, Junbin Xiao, Qingfa Xiao, Rong Xiao, Shitao Xiao, Tingsong Xiao, Tong Xiao, Yang Xiao, Yao Xiao, Yijia Xiao, Yisheng Xiao, Yongjie Xiao, Yunxuan Xiao, Yunze Xiao, Yuxin Xiao, Zhiqing Xiao, Ziang Xiao, Zilin Xiao, Chen Xiaoshu, Dong Xiaoyu, Alex Xie, Binzhu Xie, Hong Xie, Hongyan Xie, Kaige Xie, Lei Xie, Rui Xie, Ruobing Xie, Sean Xie, Sihong Xie, Tao Xie, Tianbao Xie, Wenya Xie, Xuan Xie, Yaqi Xie, Yiqing Xie, Yong Xie, Yuanzhen Xie, Yubo Xie, Yuexiang Xie, Yuqiang Xie, Yuxi Xie, Zhiwen Xie, Zhixin Xie, Zhongbin Xie, Zhongwei Xie, Zhouhang Xie, Zhuohan Xie, Amy Xin, Bowen Xing, Chen Xing, Linzi Xing, Rui Xing, Xingrun Xing, Xinyu Xing, Yue Xing, Bo Xiong, Hui Xiong, Jing

Xiong, Jinjun Xiong, Kai Xiong, Siheng Xiong, Wei Xiong, Weimin Xiong, Yun Xiong, Wei Wang Xjtlu, Baixuan Xu, Benfeng Xu, Bin Xu, Bo Xu, Boyan Xu, Cai Xu, Canwen Xu, Chejian Xu, Chen Xu, Chenchen Xu, Cong Xu, Dehong Xu, Depeng Xu, Derong Xu, Dongkuan Xu, Fangyuan Xu, Fangzhi Xu, Frank F. Xu, Hainiu Xu, Haiyang Xu, Hanzi Xu, Hao Xu, Haotian Xu, Heng-da Xu, Hong Xu, Hongshen Xu, Hongxia Xu, Hongyuan Xu, Hua Xu, Huimin Xu, Jia Xu, Jiahao Xu, Jialiang Xu, Jiarong Xu, Jin Xu, Jing Xu, Jitao Xu, Jun Xu, Justin Xu, Kaidi Xu, Kaishuai Xu, Ke Xu, Lei Xu, Lei Xu, Liang Xu, Linli Xu, Liyan Xu, Lu Xu, Lu Xu, Lvxiaowei Xu, Mayi Xu, Meilong Xu, Nan Xu, Paiheng Xu, Pengyu Xu, Qi Xu, Ran Xu, Renjing Xu, Rongwu Xu, Ruochen Xu, Shanshan Xu, Shicheng Xu, Tianyang Xu, Ting Xu, Wang Xu, Wanshi Xu, Wei Xu, Weijia Xu, Weijie Xu, Weikai Xu, Weiwen Xu, Wujiang Xu, Xi Xu, Xiangmin Xu, Xiangzhe Xu, Xiao Xu, Xin Xu, Xiongxiao Xu, Yan Xu, Yang Xu, Yang Xu, Yang Xu, Yao Xu, Yifan Xu, Yige Xu, Yongxin Xu, Yuemei Xu, Yunqiu Xu, Yuzhuang Xu, Zhangchen Xu, Zhao Xu, Zhe Xu, Zhen Xu, Zhenran Xu, Zhichao Xu, Zhijian Xu, Zhikun Xu, Zhipeng Xu, Zishan Xu, Zunnan Xu, Qi Xuan, Boyang Xue, Chun Jason Xue, Dongyu Xue, Huiyin Xue, Mingfeng Xue, Shangzi Xue, Wei Xue, Yun Xue, Zhiyu Xue, Zihan Xue, Christos Xypolopoulos

Naganand Yadati, Ajay Yadav, Prateek Yadav, Srishti Yadav, Aditya Yadavalli, Kosuke Yamada, Junichi Yamagishi, Hiroaki Yamagiwa, Atsuki Yamaguchi, Ivan P. Yamshchikov, Bing Yan, Cheng Yan, Dong Yan, Jing Nathan Yan, Junchi Yan, Junyi Yan, Ming Yan, Ruiyi Yan, Shi-qi Yan, Tianyi Yan, Weixiang Yan, Xifeng Yan, Yang Yan, Yibo Yan, Yuchen Yan, Zhaohui Yan, Zihe Yan, Bang Yang, Bohao Yang, Carl Yang, Changbing Yang, Cheng Yang, Cheng Yang, Chenxu Yang, Chih-kai Yang, Diji Yang, Dingkang Yang, Dingyi Yang, Fan Yang, Fangkai Yang, Fengyi Yang, Grace Hui Yang, Guang Yang, Guang Yang, Guanqun Yang, Haoran Yang, Hongji Yang, Ivory Yang, Jack Yang, Jiaxi Yang, Jiaxi Yang, Jingbo Yang, Jinlong Yang, Jiuding Yang, June Yong Yang, Kaicheng Yang, Kailai Yang, Kejuan Yang, Li Yang, Liang Yang, Lingyu Yang, Longfei Yang, Ming-hsuan Yang, Mu Yang, Muyun Yang, Nan Yang, Pei Yang, Qikai Yang, Qiyuan Yang, Rui Yang, Ruosong Yang, Sen Yang, Shu Yang, Shuo Yang, Sohee Yang, Soyoung Yang, Tao Yang, Tiankai Yang, Tianyi Yang, Tianyu Yang, Tianyu Yang, Wen Yang, Wenkai Yang, Wenmian Yang, Xiao-wen Yang, Xiaocheng Yang, Xiaocui Yang, Xinchen Yang, Xinyi Yang, Xinyu Yang, Xiulin Yang, Xu Yang, Yahan Yang, Yajing Yang, Yi Yang, Yi Yang, Yifan Yang, Yifan Yang, Yifan Yang, Yifei Yang, Yingxiang Yang, Yiyuan Yang, Yizhe Yang, Yongjin Yang, Yuanhang Yang, Yue Yang, Yujiu Yang, Yuming Yang, Yuqing Yang, Ze Yang, Zeyuan Yang, Zhao Yang, Zhe Yang, Zhengdong Yang, Zhenhua Yang, Zhichao Yang, Zhiyu Yang, Zhou Yang, Zi Yang, Ziqing Yang, Ziyi Yang, Roman Yangarber, Ken Yano, Binwei Yao, Chang Yao, Dingyu Yao, Duanyi Yao, Fangzhou Yao, Feng Yao, Jiashu Yao, Jihan Yao, Jing Yao, Kaisheng Yao, Liang Yao, Lili Yao, Lina Yao, Man Yao, Maxwell Yao, Peiran Yao, Quanming Yao, Wei Yao, Xingcheng Yao, Yixiang Yao, Yuekun Yao, Yuhang Yao, Yunzhi Yao, Zijun Yao, Mahsa Yarmohammadi, Michihiro Yasunaga, Mert Yazan, Bingyang Ye, Chenchen Ye, Haoran Ye, Haotian Ye, Jiabo Ye, Jiasheng Ye, Jieping Ye, Jingheng Ye, Junjie Ye, Seonghyeon Ye, Tong Ye, Wei Ye, Wentao Ye, Xiao Ye, Yang Ye, Yuyang Ye, Min-hsuan Yeh, Sung-lin Yeh, An-zi Yen, Howard Yen, Seren Yenikent, Eunjung Yeo, Gerard Christopher Yeo, Jeonghun Yeo, Antonio Jimeno Yepes, Yfguo, Euiin Yi, Jingru Yi, Jingwei Yi, Mun Yong Yi, Xiaoyuan Yi, Du Yichao, Emine Yilmaz, Jinyeong Yim, Aoxiong Yin, Congchi Yin, Dawei Yin, Hang Yin, Heqi Yin, Hongzhi Yin, Jiong Yin, Kayo Yin, Pengcheng Yin, Weichong Yin, Wenjie Yin, Xunjian Yin, Yu Yin, Yuwei Yin, Zhangyue Yin, Zhenfei Yin, Ziyi Yin, Huaiyuan Ying, Jiahao Ying, Hong Yung Yip, Siu Ming Yiu, Gal Yona, Haneul Yoo, Hee Suk Yoon, Jinsung Yoon, Seunghyun Yoon, Soyoung Yoon, Sungroh Yoon, Sunjae Yoon, Susik Yoon, Wonjin Yoon, Ori Yoran, Issei Yoshida, Ryo Yoshida, Tomoya Yoshida, Fuming You, Wangjie You, Zhiwen You, Bei Yu, Botao Yu, Dezhi Yu, Dingyao Yu, Erxin Yu, Fangxu Yu, Guoxin Yu, Hang Yu, Hang Yu, Hao Yu, Hao Yu, Haofei Yu, Hwanjo Yu, Jifan Yu, Jun Yu, Le Yu, Lequan Yu, Mengxia Yu, Nenghai Yu, Peilin Yu, Pengfei Yu, Sangwon Yu, Seunguk Yu, Shi Yu, Shoubin Yu, Sicheng Yu, Sungduk Yu, Tan Yu, Tian Yu, Tianshu Yu, Tianshu Yu, Tiezheng Yu, Weijiang

Yu, Weijie Yu, Xiao Yu, Xiaobing Yu, Xiaodong Yu, Xiaoyan Yu, Xin Yu, Xinchen Yu, Xinyan Yu, Yahan Yu, Yangyang Yu, Zeping Yu, Zhang Yu, Zhi Yu, Zhuohao Yu, Zihan Yu, Bo Yuan, Caixia Yuan, Chenhan Yuan, Chunyuan Yuan, Fei Yuan, Hongbang Yuan, Hongyi Yuan, Jiahao Yuan, Jiaqing Yuan, Jiayi Yuan, Jingyang Yuan, Li Yuan, Peiwen Yuan, Shaozu Yuan, Shuhan Yuan, Shuzhou Yuan, Siyu Yuan, Yancheng Yuan, Ye Yuan, Ye Yuan, Yige Yuan, Youliang Yuan, Zhangdie Yuan, Zheng Yuan, Zhihang Yuan, Yuanrao, Linan Yue, Murong Yue, Xiangyu Yue, Yuanhao Yue, Hyeongu Yun, Jungmin Yun, Se-young Yun, Seongjun Yun, Sukwon Yun, Taewon Yun, Tian Yun, Yunfanxie, Frances Yung, François Yvon

Joe Z, Muhammad Bilal Zafar, Hamada Zahera, Iqra Zahid, Pardis Sadat Zahraei, Mohd Abbas Zaidi, Mohammadi Zaki, Mohammed J Zaki, Mahdi Zakizadeh, Kerem Zaman, Hamed Zamani, Olga Zamaraeva, Changtong Zan, Daoguang Zan, Hongying Zan, Yuhang Zang, Andrea Zaninello, Klim Zaporojets, Alessandra Zarcone, Sina Zarrieß, Niamat Zawad, Piotr Zelasko, Aohan Zeng, Daojian Zeng, Fengzhu Zeng, Gaby Zeng, Guangtao Zeng, Hongchuan Zeng, Huimin Zeng, Jayden Zeng, Jiali Zeng, Jichuan Zeng, Jingjie Zeng, Jinshan Zeng, Min Zeng, Peijian Zeng, Qi Zeng, Qingcheng Zeng, Qingkai Zeng, Shenglai Zeng, Weixin Zeng, Xin Zeng, Yutao Zeng, Zhiyuan Zeng, Ding Zepeng, Yang Zerui, Chrysoula Zerva, George Zerveas, Albert Zeyer, Deniz Zeyrek, Sheng Zha, Zheng-jun Zha, Chengxiang Zhai, Weihe Zhai, Haolan Zhan, Hongli Zhan, Huixin Zhan, Li-ming Zhan, Pengwei Zhan, Qiusi Zhan, Runzhe Zhan, Tianyu Zhan, Zheng Zhan, Andrew Zhang, Bing Zhang, Binyan Zhang, Bo Zhang, Bohan Zhang, Bokai Zhang, Bowen Zhang, Chao Zhang, Chao Zhang, Chao Zhang, Chaoran Zhang, Chaoyun Zhang, Chen Zhang, Chenhao Zhang, Chenyang Zhang, Chi Zhang, Chong Zhang, Chong Zhang, Chunhui Zhang, Chunxia Zhang, Chuxu Zhang, Congzhi Zhang, Dong Zhang, Dongyu Zhang, Duzhen Zhang, Fan Zhang, Feng Zhang, Fengji Zhang, Fu Zhang, Fujun Zhang, Fuwei Zhang, Fuzheng Zhang, Gaifan Zhang, Ge Zhang, Gengyuan Zhang, Guchun Zhang, Haichao Zhang, Hainan Zhang, Hanlei Zhang, Hanqing Zhang, Hao Zhang, Haoran Ranran Zhang, Haowei Zhang, Haoyu Zhang, Hongbo Zhang, Hongyu Zhang, Hongzhi Zhang, Hu Zhang, Huajian Zhang, Huishuai Zhang, Huixuan Zhang, Jiajie Zhang, Jianjin Zhang, Jianshu Zhang, Jiarui Zhang, Jie Zhang, Jinchuan Zhang, Jinghao Zhang, Jinghuai Zhang, Jinghui Zhang, Jingyu Zhang, Jingyuan Zhang, Jinpeng Zhang, Jintian Zhang, Jinyi Zhang, Jipeng Zhang, Jiwen Zhang, Jizhi Zhang, Jue Zhang, Jun Zhang, Juntian Zhang, Kai Zhang, Kai Zhang, Kai Zhang, Kai Zhang, Kaike Zhang, Kaipeng Zhang, Kaiyan Zhang, Kaiyi Zhang, Ke Zhang, Ke Zhang, Kechi Zhang, Kejun Zhang, Kexun Zhang, Kun Zhang, Kun Zhang, Lan Zhang, Le Zhang, Lei Zhang, Lei Zhang, Lei Zhang, Liang Zhang, Liang Zhang, Licheng Zhang, Lin Zhang, Lingling Zhang, Lingming Zhang, Linhai Zhang, Linhao Zhang, Litian Zhang, Longhui Zhang, Longyin Zhang, Meiru Zhang, Mengqi Zhang, Mi Zhang, Mian Zhang, Michael J Zhang, Mike Zhang, Min Zhang, Mingyang Zhang, Mozhi Zhang, Muru Zhang, Peinan Zhang, Peitian Zhang, Peiyuan Zhang, Peng Zhang, Qi Zhang, Qi Zhang, Qiang Zhang, Qiang Zhang, Qihui Zhang, Qing Zhang, Qinggang Zhang, Qingjie Zhang, Qingru Zhang, Qiyuan Zhang, Richong Zhang, Rui Zhang, Ruiquan Zhang, Ruochen Zhang, Ruoyu Zhang, Sai Qian Zhang, Shao Zhang, Sheng Zhang, Shengyu Zhang, Shujian Zhang, Shuo Zhang, Sicheng Zhang, Songming Zhang, Songyang Zhang, Tao Zhang, Tianhui Zhang, Tianlin Zhang, Tianwei Zhang, Tianyang Zhang, Tianyi Zhang, Tianyi Zhang, Tianyi Zhang, Wei Zhang, Wei Emma Zhang, Weichen Zhang, Weizhi Zhang, Wen Zhang, Wen Zhang, Wenbin Zhang, Wenjun Zhang, Wenqian Zhang, Wentao Zhang, Wenyuan Zhang, Wenzheng Zhang, Xi Zhang, Xiang Zhang, Xiang Zhang, Xiangyu Zhang, Xiangyu Zhang, Xiantao Zhang, Xiao Zhang, Xiaofan Zhang, Xiaofeng Zhang, Xiaokun Zhang, Xiaoming Zhang, Xiaotong Zhang, Xiaowang Zhang, Xiaoying Zhang, Xiaoying Zhang, Xin Zhang, Xin Zhang, Xinghua Zhang, Xinjie Zhang, Xinliang Frederick Zhang, Xinlu Zhang, Xitong Zhang, Xu Zhang, Xu Zhang, Xuan Zhang, Xuan Zhang, Xuanming Zhang, Xueyao Zhang, Xuhong Zhang, Ya Zhang, Yan Zhang, Yancheng Zhang, Yanyong Zhang, Yanyue Zhang, Yanzhe Zhang, Yao Zhang, Yaping Zhang, Yazhou Zhang, Ye Zhang, Yehong Zhang, Yi Zhang, Yice Zhang, Yichao Zhang, Yichi Zhang, Yichi Zhang, Yichi Zhang, Yidan Zhang, Yifan Zhang,

Yifan Zhang, Yigeng Zhang, Yihe Zhang, Yijia Zhang, Yikai Zhang, Yiming Zhang, Yin Zhang, Ying Zhang, Yingji Zhang, Yinqi Zhang, Yiqun Zhang, Yizhuo Zhang, Yongheng Zhang, Yongqi Zhang, You Zhang, Yu Zhang, Yuan Zhang, Yuansen Zhang, Yuanxing Zhang, Yuanzhe Zhang, Yuchen Zhang, Yue Zhang, Yue Zhang, Yuejie Zhang, Yuhao Zhang, Yuhao Zhang, Yuji Zhang, Yunfan Zhang, Yunjian Zhang, Yunxiang Zhang, Yunyi Zhang, Yuqi Zhang, Yusen Zhang, Yuwei Zhang, Yuxiang Zhang, Zecheng Zhang, Zekai Zhang, Zeliang Zhang, Zequn Zhang, Zhanqiu Zhang, Zhao Zhang, Zhao Zhang, Zhehao Zhang, Zhen Zhang, Zhen Zhang, Zhen Zhang, Zhen-ru Zhang, Zheng Zhang, Zhengkun Zhang, Zhenhao Zhang, Zhenyu Zhang, Zhexin Zhang, Zheyu Zhang, Zheyuan Zhang, Zhi Zhang, Zhihan Zhang, Zhikang Zhang, Zhiyang Zhang, Zhuo Zhang, Zhuowei Zhang, Zihan Zhang, Zijian Zhang, Ziyin Zhang, Baining Zhao, Bowen Zhao, Chenxi Zhao, Fei Zhao, Feng Zhao, Feng Zhao, Guangxiang Zhao, Guangzhen Zhao, Guoshuai Zhao, Haiquan Zhao, Haiteng Zhao, Haiyan Zhao, Hengyuan Zhao, Hui Zhao, Jiahao Zhao, Jianqiao Zhao, Jianyu Zhao, Jiaxu Zhao, Jie Zhao, Jin Zhao, Jinman Zhao, Junbo Zhao, Kai Zhao, Kaiyan Zhao, Kun Zhao, Libo Zhao, Lili Zhao, Lingchen Zhao, Lingjun Zhao, Lulu Zhao, Mengjie Zhao, Mengyu Zhao, Minyi Zhao, Pengyu Zhao, Pu Zhao, Qinghua Zhao, Rui Zhao, Ruining Zhao, Runhao Zhao, Ruochen Zhao, Sendong Zhao, Shiwan Zhao, Shu Zhao, Shuai Zhao, Wei Zhao, Weilin Zhao, Weixiang Zhao, Wenlong Zhao, Wenting Zhao, Xiangyu Zhao, Xiangyu Zhao, Xiaoyan Zhao, Xingmeng Zhao, Xingyi Zhao, Xinping Zhao, Xinyu Zhao, Xiutian Zhao, Xuechen Zhao, Xueliang Zhao, Xujiang Zhao, Yang Zhao, Yang Yang Zhao, Yanyan Zhao, Yi Zhao, Yibo Zhao, Yiran Zhao, Yiren Zhao, Yizhou Zhao, Yongwei Zhao, Yu Zhao, Yu Zhao, Yukun Zhao, Yunxiao Zhao, Yutian Zhao, Yuze Zhao, Zhe Zhao, Zheng Zhao, Zhengyu Zhao, Zhenjie Zhao, Zhou Zhao, Zihao Zhao, Zijian Zhao, Zirui Zhao, Zhaolingchen, Bi Zhen, Ranran Zhen, Changmeng Zheng, Duo Zheng, Guangmin Zheng, Huan Zheng, Huanran Zheng, Jingnan Zheng, Jonathan Zheng, Kai Zheng, Mengxin Zheng, Mingyu Zheng, Shen Zheng, Tianshi Zheng, Tianyu Zheng, Tong Zheng, Weiguo Zheng, Weixiong Zheng, Xiaolin Zheng, Xiaoqing Zheng, Xiawu Zheng, Yefeng Zheng, Yi Zheng, Yilun Zheng, Yinhe Zheng, Yongsen Zheng, Zhe Zheng, Zhi Zheng, Zi'ou Zheng, Zibin Zheng, Zihao Zheng, Shi Zhi, Zhibinlan, Jialun Zhong, Jianyuan Zhong, Liangjun Zhong, Meizhi Zhong, Mian Zhong, Qihuang Zhong, Shanshan Zhong, Shaochen Zhong, Xian Zhong, Yang Zhong, Yujie Zhong, Zhi Zhong, Baohang Zhou, Bin Zhou, Changzhi Zhou, Cheng-jie Zhou, Chuang Zhou, Dongzhan Zhou, Fan Zhou, Gengze Zhou, Hao Zhou, Hao Zhou, Houquan Zhou, Huachi Zhou, Huichi Zhou, Huiwei Zhou, Jiaming Zhou, Jiayu Zhou, Jinfeng Zhou, Junsheng Zhou, Kang Zhou, Kun Zhou, Lexin Zhou, Li Zhou, Linqi Zhou, Luowei Zhou, Meng Zhou, Mengyu Zhou, Mingyang Zhou, Mingyao Zhou, Naitian Zhou, Peilin Zhou, Qiji Zhou, Shijia Zhou, Shilin Zhou, Shuigeng Zhou, Tianshuo Zhou, Tianyi Zhou, Tong Zhou, Wei Zhou, Wei Zhou, Weixiao Zhou, Wengang Zhou, Xiabing Zhou, Xiangyang Zhou, Xiaobing Zhou, Xiaoling Zhou, Xin Zhou, Xin Zhou, Xinyu Zhou, Yan Zhou, Yang Zhou, Yang Zhou, Yaqian Zhou, Yijie Zhou, Yin Zhou, Ying Zhou, Yongxin Zhou, Yougen Zhou, Yue Zhou, Yufa Zhou, Yufan Zhou, Yuhang Zhou, Yujun Zhou, Yukai Zhou, Yunhua Zhou, Yuqi Zhou, Yuxiang Zhou, Zhanke Zhou, Zhejian Zhou, Zhenhong Zhou, Zhoujunhao, Bolin Zhu, Conghui Zhu, Dawei Zhu, Derui Zhu, Fangqi Zhu, Fangwei Zhu, Hanlun Zhu, Henghui Zhu, Hongsong Zhu, Jia Zhu, Jian Zhu, Jianke Zhu, Jing Zhu, Junda Zhu, Junnan Zhu, Junyi Zhu, Kaijie Zhu, Kenny Q. Zhu, Kun Zhu, Linchao Zhu, Luyao Zhu, Mengna Zhu, Minfeng Zhu, Ming Zhu, Minjun Zhu, Muhua Zhu, Peide Zhu, Qi Zhu, Qingfu Zhu, Rui-jie Zhu, Shanfeng Zhu, Shaolin Zhu, Song-chun Zhu, Tinghui Zhu, Tong Zhu, Wang Zhu, Wei Zhu, Xiangrong Zhu, Xiaofeng Zhu, Xiaofeng Zhu, Xuekai Zhu, Yaochen Zhu, Yeshuang Zhu, Yi Zhu, Yi Zhu, Yingjie Zhu, Yinglun Zhu, Yonghua Zhu, Yongxin Zhu, Yutao Zhu, Yuying Zhu, Zhihong Zhu, Ziwei Zhu, Zixiao Zhu, Fuzhen Zhuang, Haojie Zhuang, Haomin Zhuang, Huiping Zhuang, Shengyao Zhuang, Xiang Zhuang, Xianwei Zhuang, Xinlin Zhuang, Yuan Zhuang, Yuchen Zhuang, Ziyu Zhuang, Jingming Zhuo, Alireza Salkhordeh Ziabari, Caleb Ziems, Brian Douglas Zimmerman, Roger Zimmermann, Yftah Ziser, Xu Zixiang, Zixuanwang, Stefan Zohren, Chang Zong, Linlin Zong, Qing Zong, Zongke, Bocheng Zou, Henry Peng Zou, Heqing Zou, Jiaru Zou, Jie Zou, Lixin Zou, Yanyan Zou, Zhuoyang Zou, Amal Zouaq, Vilém

Zouhar, Andrea Zugarini, Liang Zujie, Jialung Zuo, Kaiwen Zuo, Max Zuo, Yuxin Zuo

Keynote Talk

Rethinking Pretraining: Data and Architecture

Luke Zettlemoyer
Paul G. Allen School of Computer Science and Engineering, University of Washington

Mon, July 28th, 2025 – Time: **09:30 – 10:30** – Room: **Austria Center Vienna**

Abstract: Large language model training follows a standard pipeline: tokenization, pretraining, possibly mid-training, and post training or alignment. Despite its wild success, we understand relatively little about this recipe and are almost certainly missing many opportunities to improve it. In this talk, I will focus on three such cases. I'll describe our work on data efficient post training (e.g. LIMA, ALMA, and s1) where we argue that nearly all advanced model capabilities ultimately come from the pretraining data, even if effective alignment is still essential for controlling model behavior. I will also describe new methods for extracting more signal from the pretraining data, including new hierarchical architectures for byte-level language models (e.g. BLT) that are both tokenizer-free and scale better than traditional BPE-based methods, especially in the long tail. Finally, I will discuss decentralized, modular training algorithms (e.g. BTM) that better isolate and control the influence of specific data on specific model components and behaviors. Together, these methods promise to simplify training and improve scaling, by centering and amplifying the influence of data in architecture design.

Bio: Luke Zettlemoyer is a Professor in the Paul G. Allen School of Computer Science & Engineering at the University of Washington, and a Senior Research Director at Meta. His research focuses on empirical methods for natural language semantics, and involves designing machine learning algorithms, introducing new tasks and datasets, and, most recently, studying how to best develop new architectures and self-supervision signals for pre-training. His honors include being elected ACL President, named an ACL Fellow, winning a PECASE award, an Allen Distinguished Investigator award, and multiple best paper awards. Luke was an undergrad at NC State, received his PhD from MIT and was a postdoc at the University of Edinburgh.

Keynote Talk

Whose Gold? Re-imagining Alignment for Truly Beneficial AI

Verena Rieser
Google DeepMind

Tue, July 29th, 2025 – Time: **09:00 – 10:00** – Room: **Austria Center Vienna**

Abstract: Human feedback is often the "gold standard" for AI alignment, but what if this "gold" reflects diverse, even contradictory human values? This keynote explores the technical and ethical challenges of building beneficial AI when values conflict – not just between individuals, but also within them. My talk advocates for a dual expansion of the AI alignment framework: moving beyond a single, monolithic viewpoint to a plurality of perspectives, and transcending narrow safety and engagement metrics to promote comprehensive human well-being.

Bio: Verena Rieser is a Senior Staff Research Scientist at Google DeepMind, where she founded the VOICES team (Voices-of-all in alignment). Her team is a core contributor to Gemini with a mission to enhance model safety and usability for diverse communities. Verena has pioneered work in data-driven multimodal Dialogue Systems and Natural Language Generation, encompassing conversational RL agents, faithful data-to-text generation, spoken language understanding, evaluation methodologies, and applications of AI for societal good. Verena previously directed the NLP lab as a full professor at Heriot-Watt University, Edinburgh, and held a Royal Society Leverhulme Senior Research Fellowship. She earned her PhD from Saarland University.

Panel
Generalization of NLP Models

Mon, July 28th, 2025 – Time: **16:30 – 17:30** – Room: **Austria Center Vienna - Level 2 Hall A - C**

Moderator: Eduard Hovy, Melbourne Connect University of Melbourne

Eduard Hovy is the Executive Director of Melbourne Connect (a research and tech transfer centre at the University of Melbourne), a professor at the University of Melbourne's School of Computing and Information Systems, and an adjunct professor at the Language Technologies Institute in the School of Computer Science at Carnegie Mellon University. In 2020–21 he served as Program Manager in DARPA's Information Innovation Office (I2O), where he managed programs in Natural Language Technology and Data Analytics. Dr. Hovy completed a Ph.D. in Computer Science (Artificial Intelligence) at Yale University in 1987 and was awarded honorary doctorates from the National Distance Education University (UNED) in Madrid in 2013 and the University of Antwerp in 2015. He is one of the initial 17 Fellows of the Association for Computational Linguistics (ACL) and is also a Fellow of the Association for the Advancement of Artificial Intelligence (AAAI). Dr. Hovy's research focuses on computational semantics of language and addresses various areas in Natural Language Processing and Data Analytics, including in-depth machine reading of text, information extraction, automated text summarization, question answering, the semi-automated construction of large lexicons and ontologies, and machine translation. In early 2025 his Google h-index was 109, with about 68,000 citations. Dr. Hovy is the author or co-editor of eight books and over 400 technical articles and is a popular invited speaker. He regularly co-taught Ph.D.-level courses and has served on Advisory and Review Boards for both research institutes and funding organizations in Germany, Italy, Netherlands, Ireland, Singapore, and the USA. From 2003 to 2015 he was co-Director of Research for the Department of Homeland Security's Center of Excellence for Command, Control, and Interoperability Data Analytics, a distributed cooperation of 17 universities. In 2001 Dr. Hovy served as President of the international Association of Computational Linguistics (ACL), in 2001–03 as President of the International Association of Machine Translation (IAMT), and in 2010–11 as President of the Digital Government Society (DGS).

Panelists:

Mirella Lapata, University of Edinburgh

Mirella Lapata is professor of natural language processing in the School of Informatics at the University of Edinburgh. Her research focuses on getting computers to understand, reason with, and generate natural language. She is the first recipient (2009) of the British Computer Society and Information Retrieval Specialist Group (BCS/IRSG) Karen Sparck Jones award and a Fellow of the Royal Society of Edinburgh, the ACL, and Academia Europaea. Mirella has also received best paper awards in leading NLP conferences and has served on the editorial boards of the Journal of Artificial Intelligence Research, the Transactions of the ACL, and Computational Linguistics. She was president of SIGDAT (the group that organizes EMNLP) in 2018. She has been awarded an ERC consolidator grant, a Royal Society Wolfson Research Merit Award, and a UKRI Turing AI World-Leading Researcher Fellowship.

Dan Roth, University of Pennsylvania and Oracle

Dan Roth is the Eduardo D. Glandt Distinguished Professor at the University of Pennsylvania and Chief AI Scientist at Oracle. Until June 2024 Dan was a VP/Distinguished Scientist at AWS AI where he led the scientific effort behind Amazon's first-generation GenAI products, including Titan Models, Amazon Q, and Amazon Bedrock. Dan is a Fellow of the AAAS, ACM, AAAI, and ACL, and a recipient of the IJCAI John McCarthy Award "for major conceptual and theoretical advances in the modeling of natural language understanding, machine learning, and reasoning." He has published broadly in natural language processing, machine learning, knowledge representation and reasoning, and learning theory, was the Editor-in-Chief of the Journal of Artificial Intelligence Research (JAIR) and has served as a Program Chair and Conference Chair for the major conferences in his research areas. Roth has been involved in several ML/NLP/GenAI startups in domains that range from legal and compliance to health care. Dan received his B.A Summa cum laude in Mathematics from the Technion, Israel and his Ph.D. in Computer Science from Harvard University in 1995.

Yue Zhang, Westlake University

Yue Zhang is a tenured Professor at Westlake University (https://frcchang.github.io). His research interests include fundamental NLP and its machine learning algorithms, and his recent research focuses on LLM reasoning and AI scientist. His major contributions to the field include machine learning algorithms for structured prediction (e.g., parsing and IE), neural NLP models (i.e., lattice and graph LSTM), and generalization for NLP/LM (e.g., OOD and logical reasoning). He co-authored the Cambridge University Press book "Natural Language Processing – a Machine Learning Perspective" and served as a PC co-chair for CCL 2020 and EMNLP 2022, test-of-time award committee co-chairs for ACL 2024 and 2025, action editor for TACL, and associate editor for TASLP, TALLIP, TBD, and CSL. He won the best paper awards of IALP 2017 and COLING 2018, best paper honorable mention of SemEval 2020, and best paper nomination for ACL 2018 and ACL 2023.

Table of Contents

Towards LLM-powered Attentive Listener: A Pragmatic Approach through Quantity Self-Repair

Junlin Li[1], Bo Peng[1], Yu-yin Hsu[1],
[1]Department of Chinese and Bilingual Studies, The Hong Kong Polytechnic University (PolyU)
[1]junlin.li@connect.polyu.hk

Abstract

Grice's Quantity Maxims dictate that human speakers aim for the optimal quantity of information during conversation. To empower LLMs to self-repair their responses toward optimal quantity and improve their attentive listening skills, we propose **Q-Tuning** and **Q-Traveling**, which draw on heuristic pathfinding to enable decoder-only LLMs to travel among multiple "Q-alternatives" (Quantity Alternatives) and search for the optimal quantity in coordination with a conversation goal. Automatic and human evaluations demonstrate the effectiveness of Q-Tuning and Q-Traveling in constructing human-like, user-centered conversation agents. Our repository is open-sourced via https://github.com/CN-Eyetk/QTraveling.

1 Introduction

Quote to Dorothy Nevill, "*the real art of conversation is not only to say the right thing at the right place but to leave unsaid the wrong thing at the tempting moment*" (Nevill, 1910).

To hold back the wrong thing from being said, people pay attention to their addressees' expectations and self-repair their inner speech before speaking (Levelt, 1983). As illustrated in Figure 1, this pragmatic wisdom is overtly reflected in self-repair practices that are productive in real-world conversations (Sun, 2022), especially in attentive listening (Sarira et al., 2023). The self-repair strategies reflect listeners' attention to their addressees' expectations generated by various conversation principles, typically the Cooperative Principle (Good, 1990).

Taking QUANTITY MAXIMS as an instance, attentive listeners tactfully pursue an optimal quantity of information to achieve their conversation goals (Hossain et al., 2021; Atifi et al., 2011). As illustrated in Figure 1 attentive listeners should monitor and repair the quantity (or informativeness) of their utterances to achieve the optimal communicative effect. Being over-informative or under-informative violates the QUANTITY MAXIMS and thus yields non-literal meaning and pragmatic failure (Blum-Kulka and Olshtain, 1986).

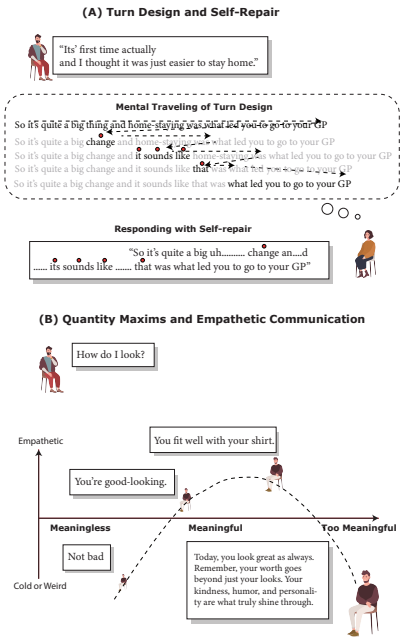


Figure 1: Self-Repair Pracitces and Quantity Maxims: People aim for optimal quantity through self-repair.

Despite their advancement, it is still questionable whether the decoder-only LLMs, as empathetic listeners, are human-like and attentive in essence (Pan et al.; Cuadra et al., 2024; Yin et al., 2024). LLM responses are perceived as hollow (Yin et al., 2024) and insincere (Lee et al., 2024), with limited attention to exploring and interpreting the user's experience (Cuadra et al., 2024). This pitfall presumably reflects the drawback of incremental language generation in manipulating the quantity of their response, which is an important conversation strategy (Yeung et al., 1999).

To address this human-model misalignment, we propose theory-driven tuning and language genera-

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1–13
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

tion paradigms, **Q-Tuning** and **Q-Traveling**, to improve LLM's attentive listening skills through the "covert" self-repair process that frequently occurs in real-world communication. Narrowing down upon Grice's QUANTITY MAXIMS, we tune a pre-trained LM to explore multiple Quantity Alternatives. During inference, we inform the pragmatic-aware LM to search for the optimal "Q-alternative" (quantity alternative) among the alternatives in pursuit of a flexible scoring function. Following the A* search algorithm (Hart et al., 1968), an optimal Q-alternative grounded in heuristics can be written after a chain of self-repair operations.

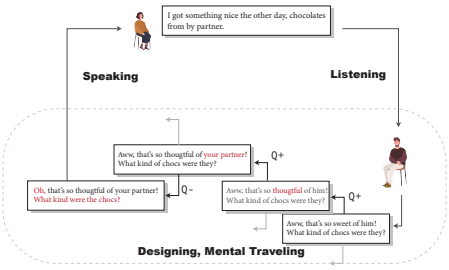


Figure 2: Generating Attentive Response through Traveling among "Q-alternatives" ("Q+" for providing more specific information, 'Q-" for providing less specific information)

Of sufficient relevance to our study are the post hoc correction or self-correction methods. (Kim et al., 2024; Madaan et al., 2024; Qu et al., 2024) Distinct from the RL (Reinforcement-learning) methods based on a static reward function, the current study proposes a novel and plug-and-play self-correction paradigm based on a controllable heuristic goal. The Q-Traveling method improves the contextual adaptability to variable needs and desires in real-world users. It also presents an operationalizable framework to incorporate implicit linguistic-pragmatic knowledge, typically Grice's Maxims of Conversation, into LLM-powered dialogue systems.

The major contributions of this study include:

- We propose **Q-Tuning** to infuse Quantity Maixms into LLMs. The evaluation results demonstrate a decisive contribution of this tuning paradigm to empathetic and attentive listening skills.
- We propose **Q-Traveling** to plan out the optimal pragmatic alternative through seir-repair path-finding. Drawing on the A* search algorithm, Q-Traveling seamlessly guides LLM listeners to an adaptable scoring function, improving LLM listeners' competence to deal with versatile conversation goals.

2 Preliminary

2.1 Quantity Maxims

QUANTITY MAXIMS consist of a lower-bound maxim and an upper-bound maxim (Grice, 1975; Carston, 1995):

- MAXIM-I: *Make your contribution as informative as is required (for the current purposes of the exchange).*
- MAXIM-II: *Do not make your contribution more informative than is necessary*

Consider the operationalization of QUANTITY MAXIMS in a dialogue system, for a set of unidirectionally-entailing utterances as Q-alternatives $U = \{u_1, u_2, \cdots, u_{n-1}, u_n\}$ where $u_n \vDash u_{n-1} \vDash \cdots \vDash u_2 \vDash u_1$ [1], there exists an "optimal" alternative (at least "good enough") u_* in context C given a heuristic function $\mathcal{H}$.

$$u_* = \arg\max_{u \in U} \ \|\mathcal{H}(u|C)\|. \tag{1}$$

2.2 Problem Formulation - Optimal Quantity Alternative

The conventional practice of dialogic systems requires a language model $\mathcal{M}_\theta$ to generate a response u_0 from the dialogic context C.

$$u^0 \sim \mathcal{M}_\theta(C) \tag{2}$$

To search for the optimal Q-alternative, we induce $\mathcal{M}_\theta$ to conduct a pair of Quantity Guidances $q \in \{Q^+, Q^-\}$, where Q^+ denotes providing more specific information (following MAXIM-I) and Q^- denotes providing less specific information (following MAXIM-II). We expect the model to iteratively repair its current response to include more information (when $q = Q^+$, so that $u^t \vDash u^{t-1}$) or include less information (when $q = Q^-$, so that $u^{t-1} \vDash u^t$).

$$u^t \sim \mathcal{M}_\theta(u^{t-1}|q, C). \tag{3}$$

To achieve goal-driven self-repair, we use a heuristic function $\mathcal{H}$ to explore the optimal repair path

[1] We use $v \vDash$ to denote semantic entailment.

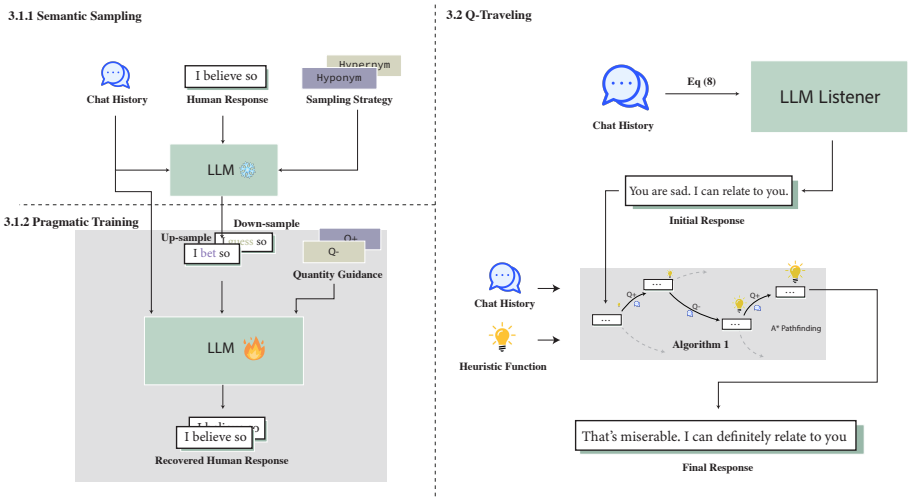


Figure 3: The overview of our method. Q-Tuning draws on the model's inner semantic knowledge to train pragmatic strategies. Q-Traveling instructs the model to explore and search out the optimal Q-alternative.

$\{u^0 \xrightarrow{q^0} u^1 \xrightarrow{q^1} \cdots u^T\}$, so that u^T is the optimal alternative of u^0.

$$u^T = \underset{u \sim \mathcal{M}_\theta(u^0)}{\arg\max} \; \|\mathcal{H}(u|C)\|. \tag{4}$$

3 Method

Human interlocutors, with a set of Q-alternatives in mind, design their turns to conform with Grice's maxims. Inspired by this process, we propose the tuning and inference paradigm in the following.

3.1 Quantity Maxims Tuning (Q-Tuning)

We initially equip a pre-trained LLM with the pragmatic knowledge to repair an utterance according to a given Quantity Guidance $q \in \{Q^+, Q^-\}$. To train this ability, we leverage the LLM's prior semantic knowledge to create paired training samples with minimal semantic contrasts.

3.1.1 Semantic Sampling for Minimal Pairs

Given a human annotation u^h, we prompt a pre-trained LLM $\mathcal{M}_\theta$ to get a down-sample u^{h-} and an up-sample u^{h+}. We use the strategies in the following to control the semantic relation between the source, up and down samples.

- To obtain u^{h^-}, $\mathcal{M}_\theta$ is asked to (1) substitute a word or phrase with its hypernym expression or (2) remove a word or phrase.
- To obtain u^{h^+}, $\mathcal{M}_\theta$ is asked to (1) substitute a word or phrase with its hyponym expression or (2) include a word or phrase.

We add two constraints to the prompt as follows:

- u^{h^-} should be semantically entailed by u^h, and u^{h^-} should be congruous with the context C
- u^{h^+} should semantically entail u^h, and u^{h^+} should be congruous with the context C

Details of prompting and quality check are in the Appendix C.

3.1.2 Pragmatic Training for Quantity Self-Repair

To train self-repair behavior, we treat each u^h as the label and its corresponding u^{h^-} and u^{h^+} as input. The training loss can be formulated as below:

$$\mathcal{L}^+ = -\sum_{j=1}^{|u^h|} log\mathcal{M}_{\theta,\alpha}\left(u^h_j | u^h_{<j}, u^{h^-}, Q^+, C\right) \tag{5}$$

$$\mathcal{L}^- = -\sum_{j=1}^{|u^h|} log\mathcal{M}_{\theta,\alpha}\left(u^h_j | u^h_{<j}, u^{h^+}, Q^-, C\right) \tag{6}$$

$$\mathcal{L} = \mathcal{L}^+ + \mathcal{L}^- \tag{7}$$

where α denotes the adapter subnetwork injected during adapter tuning.

3.2 Response Initializing

We find that the post-trained model $\mathcal{M}_{\theta,\alpha}$ is still able to generate an initial response from scratch.

$$u_0 \sim \mathcal{M}_{\theta,\alpha}(C) \tag{8}$$

3.3 Inter-Quantity Traveling (Q-Traveling)

We propose **Q-Traveling** to search for the optimal Q alternative based on a scoring function $\mathcal{H}(u)$.

Algorithm 1 Heuristic Search for Optimal Quantity

Input: $u_0, c, \mathcal{M}_{\theta,\alpha}, \mathcal{H}$
  $\texttt{open} \leftarrow [u_0]$
  $\texttt{close} \leftarrow \emptyset$
  $\texttt{score} \leftarrow \{\}, \texttt{score}[u_0] \leftarrow \mathcal{H}(u_0)$
  while $\texttt{open} \neq \emptyset \ \& \ |\texttt{close}| <= \texttt{maxstep}$ **do**
    $\texttt{open} \leftarrow \underset{u \in \texttt{open}}{\text{argsort}}(\texttt{score}(u))$
    $u^p \leftarrow \texttt{pop}(\texttt{open})$
    $u^{p^+} = \texttt{generate}(\mathcal{M}_{\theta,\alpha}, Q^+, u^p)$
    $u^{p^-} = \texttt{generate}(\mathcal{M}_{\theta,\alpha}, Q^-, u^p)$
    for $u \in \left[u^{p^+}, u^{p^-}\right]$ **do**
      $\texttt{score}[u] = \mathcal{H}(u)$
      $\texttt{append}(\texttt{open}, u)$
    end for
    $\texttt{append}(\texttt{close}, u^{'})$
  end while
Output: $u^* \leftarrow \underset{u \in \texttt{score.keys}}{\text{argmax}} \ (\texttt{score}(u))$

The heuristic search algorithm is presented in Algorithm 1. In each iteration, according to the scoring board `score`, we sort the open list `open` in descending order and pop the first response as the parent node u^p. We extend two new responses u^{p^+} and u^{p^-} by implementing Q^+ and Q^-. We score the two new responses with $\mathcal{H}$ and register the scores on the scoring board. We append the two new responses to the open set at the end of the iteration. We terminate the iteration when the maximum number of extended responses has been reached. Finally, we select the response with the highest score from the scoring board.

4 Experiments

LlaMA+Q-Traveling v.s. LlaMA	win	lose	tie
Human-like	**41.7**†	30	28.3
Empathetic	**41.0**†	32.3	26.7
Attentive	**46.7**†	40	13.3

Table 2: Results of Human Evaluation. †denotes a significant improvement of $p < 0.05$.

We implemented experiments in two data sets: EMPATHETICDIALOGUE (ED) and EMOTIONAL-SUPPORT-CONVERSATION (ESC). Implementation details and baselines are described in Appendix A.

Above traditional rule-based metrics such as distinct score (**Dist**) and bleu score (**BLEU**), we also pay attention to model-based metrics such as **AI-rate**, expected judgement about empathy (**EmotionalReactions**, **Interpretation**, **Exploration**) based on the framework of EPITOME (Sharma et al., 2020), as well as the similarity between the output of the system and the ground truth in terms of emotion (**SimEMO**) and personality (**SimPerson**).

As shown in Table 1, our method leads to a visible increase in system performance in terms of human-like and diverse language use. Inspecting both data sets, the Q-Tuning and Q-Traveling mechanism also enlarges the diversity (**Dist-n**). The reduction in the use of AI-like language use (**AI-rate**) is also noticeable compared to LLM baselines. We also observe an improvement in the match of emotion and personality (**SimEMO** and **SimPerson**) with ground truth, mostly owing to Q-Tuning.

Table 2 presents the results of the human evaluations. Our approach shows a remarkable advantage with respect to the use of human-like and attentive language.

5 Analysis

Figure 4 compares the distribution of personality embeddings (see A.5) from the LLM backbone, our repair-aware systems, and human-written ground truth. With the proposed mechanism for quantity repair, the system output is densely distributed in a human-like subzone (marked in a red oval), compared to backbone LLMs.

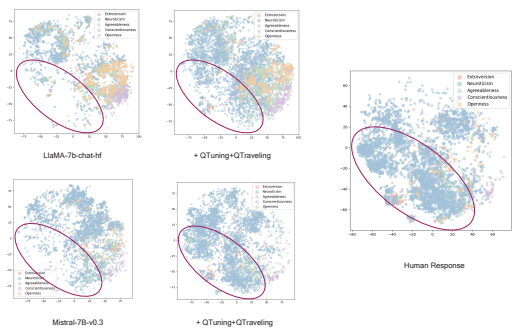


Figure 4: Q-Tuning and Q-Traveling anchor the personality embeddings to a more human-like subzone

We also inspect two different goals, including (1) empathetic reaction and (2) helpfulness and

Dataset	Model	Dist-1	Dist-3	BLEU-1	AI-rate	SimEMO	SimPerson	EmotionalReaction	Interpretation	Exploration
ED	CARE	0.63	3.89	20.02	63.56	36.73	81.11	1.15	0.02	0.58
	SEEK	0.62	4.09	9.54	61.00	41.36	80.23	0.35	0.14	0.29
	LLaMA	3.55	48.59	12.33	70.64	54.47	76.24	0.88	0.10	**0.82**
	+QTune	**3.99**	44.86	**15.57**	66.45	**54.74**	**80.07**	**0.99**	**0.12**	0.68
	+QTune+QTravel	3.97	**49.24**	14.53	**65.19**	54.14	79.06	0.98	0.12	0.67
	Mistral	3.68	48.21	15.32	71.07	54.01	80.38	**0.97**	0.08	0.62
	+QTune	4.50	49.37	**20.25**	65.19	**55.83**	**82.40**	0.87	**0.19**	0.99
	+QTune+QTravel	**4.61**	**55.51**	17.58	**58.95**	55.20	81.06	0.85	0.16	**1.09**
ESC	VLESA	3.19	33.43	23.54	65.17	52.00	79.86	1.02	0.70	0.41
	Cooper	4.16	33.33	22.00	66.01	50.27	80.30	0.98	0.62	0.33
	LLaMA	**5.00**	49.93	17.30	71.24	51.00	77.58	0.84	0.10	0.56
	+QTune	4.69	52.15	**18.87**	**63.90**	**52.37**	**78.16**	**0.96**	**0.13**	0.60
	+QTune+QTravel	4.74	**58.83**	15.40	64.84	51.21	76.09	0.89	0.11	0.56
	Mistral	3.68	48.21	15.32	69.01	53.45	**79.08**	**0.97**	0.09	0.66
	+QTune	**4.65**	61.06	**19.46**	58.59	**53.60**	78.76	0.84	**0.15**	1.04
	+QTune+QTravel	4.30	**63.68**	15.39	**57.20**	52.46	77.44	0.81	0.11	**1.10**

Table 1: Results of Automatic Evaluation. Best performance among LLM-powered and among all systems are bold-highlighted and underlined seperately.

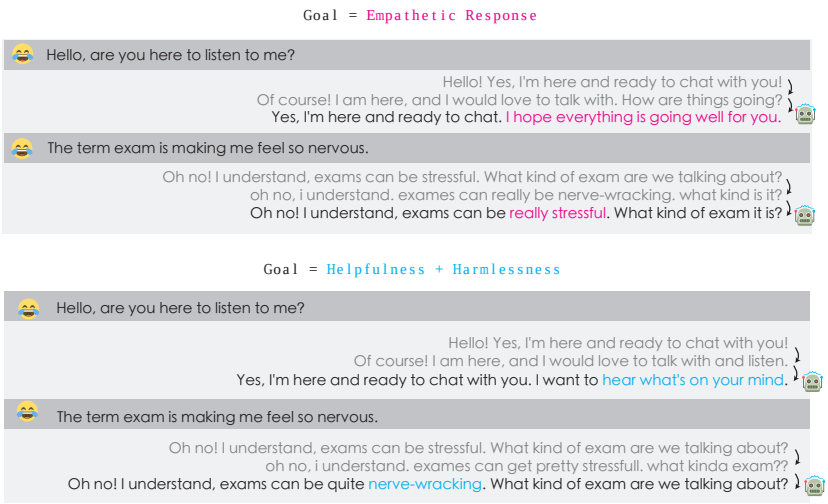


Figure 5: Q-Traveling reflects goal-driven conversation: the effect of scoring function on lexical choice.

harmlessness (see A.4). From the case presented in Figure 5, we notice the adaptability of our system to different conversation goals. Detailed case studies are given in the Appendix B.1

6 Conclusions

Inspired by quantity self-repair practices in real-world conversation, we propose **Q-Tuning** and **Q-Traveling** to infuse pragmatic conversation strategies into large language models. The results indicate a noticeable improvement in human-like attentive listening skills.

Limitations

The paper focuses primarily on the impact of Quantity Maxims on human conversation without delving into potential cultural or situational factors that might influence these dynamics. The study may not account for individual differences in how different listeners interpret and respond to varying levels of informativeness, which could limit the generalizability of the findings.

Finally, assessing the precise impact of conversation maxims on empathy and mental health outcomes could be challenging due to the subjective nature of these constructs and the difficulty in quantifying such effects accurately. More subjective judgment data should be collected and annotated to provide a solution to the issue under discussion.

Ethical Considerations

Our study is based on the ESC and ED dataset, designed specifically for emotional support and empathetic conversations and openly available for research purposes. These data sets maintain a focus on empathy-driven scenarios while ensuring the exclusion of sensitive or personal data and unethical

language. Throughout our research, the utmost priority was given to safeguarding the privacy of all participants involved.

It is also crucial to clarify that our dialogue system is not intended to address or improve outcomes in high-risk or nonroutine scenarios such as those involving self-harm or suicide. We recognize the indispensable role of professional psychological counseling or treatment in managing such critical situations.

Finally, all human participants involved in the evaluation process provide informed consent. To maintain the confidentiality and anonymity of participants, all human evaluation data was handled with strict confidentiality measures in place. The whole human-recruiting procedures are approved by The PolyU Institutional Review Board (IRB).

Acknowledgments

This research was supported by GRF (B-Q0AJ), and by the Hong Kong Polytechnic University through the Large Equipment Fund (1-BC7N), CBS fund (1-W16H). We would also like to thank the anonymous reviewers for their feedback and suggestions.

References

Hassan Atifi, Sacha Mandelcwajg, and Michel Marcoccia. 2011. The co-operative principle and computer-mediated communication: The maxim of quantity in newsgroup discussions. *Language Sciences*, 33(2):330–340.

Shoshana Blum-Kulka and Elite Olshtain. 1986. Too many words: Length of utterance and pragmatic failure. *Studies in second language acquisition*, 8(2):165–179.

Robyn Carston. 1995. Quantity maxims and generalised implicature. *Lingua*, 96(4):213–244.

Yi Cheng, Wenge Liu, Wenjie Li, Jiashuo Wang, Ruihui Zhao, Bang Liu, Xiaodan Liang, and Yefeng Zheng. 2022. Improving multi-turn emotional support dialogue generation with lookahead strategy planning. *arXiv preprint arXiv:2210.04242*.

Yi Cheng, Wenge Liu, Jian Wang, Chak Tou Leong, Yi Ouyang, Wenjie Li, Xian Wu, and Yefeng Zheng. 2024. Cooper: Coordinating specialized agents towards a complex dialogue goal. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17853–17861.

Andrea Cuadra, Maria Wang, Lynn Andrea Stein, Malte F Jung, Nicola Dell, Deborah Estrin, and James A Landay. 2024. The illusion of empathy? notes on displays of emotion in human-computer interaction. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–18.

David Good. 1990. Repair and cooperation in conversation. In *Computers and conversation*, pages 133–150. Elsevier.

Herbert Paul Grice. 1975. Logic and conversation. *Syntax and semantics*, 3:43–58.

Peter E Hart, Nils J Nilsson, and Bertram Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107.

Md Mahroof Hossain et al. 2021. The application of grice maxims in conversation: A pragmatic study. *Journal of English Language Teaching and Applied Linguistics*, 3(10):32–40.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2024. Language models can solve computer tasks. *Advances in Neural Information Processing Systems*, 36.

Yoon Kyung Lee, Jina Suh, Hongli Zhan, Junyi Jessy Li, and Desmond C Ong. 2024. Large language models produce responses perceived to be empathic. *arXiv preprint arXiv:2403.18148*.

Willem JM Levelt. 1983. Monitoring and self-repair in speech. *Cognition*, 14(1):41–104.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Junlin Li, Bo Peng, Yu-Yin Hsu, and Chu-Ren Huang. 2024. Be helpful but don't talk too much-enhancing helpfulness in conversations through relevance in multi-turn emotional support. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1976–1988.

Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang, and Minlie Huang. 2021. Towards emotional support dialog systems. *arXiv preprint arXiv:2106.01144*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Lady Dorothy Nevill. 1910. *Under Five Reigns*. Methuen.

Siyu Pan, Caoyun Fan, Binglei Zhao, Siyang Luoc, and Yaohui Jin. Can large language models exhibit cognitive and affective empathy as humans?

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. Recursive introspection: Teaching language model agents how to self-improve. *arXiv preprint arXiv:2407.18219*.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2018. Towards empathetic open-domain conversation models: A new benchmark and dataset. *arXiv preprint arXiv:1811.00207*.

Pret Sarira, Murni Mahmud, Akhmad Affandi, and Muftihaturrahmah Burhamzah. 2023. The existence of fillers in converting the written language to spoken language. *Borneo Journal of English Language Education*, 5(1).

Ashish Sharma, Adam Miner, David Atkins, and Tim Althoff. 2020. A computational approach to understanding empathy expressed in text-based mental health support. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5263–5276, Online. Association for Computational Linguistics.

Xiyue Sun. 2022. A corpus based study on self-repairs in chinese english learners' oral production. In *2021 International Conference on Education, Language and Art (ICELA 2021)*, pages 539–543. Atlantis Press.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jiashuo Wang, Yi Cheng, and Wenjie Li. 2022a. CARE: Causality reasoning for empathetic responses by conditional graph generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 729–741, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lanrui Wang, Jiangnan Li, Zheng Lin, Fandong Meng, Chenxu Yang, Weiping Wang, and Jie Zhou. 2022b. Empathetic dialogue generation via sensitive emotion recognition and sensible knowledge selection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4634–4645, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lorrita NT Yeung, Timothy R Levine, and Kazuo Nishiyama. 1999. Information manipulation theory and perceptions of deception in hong kong. *Communication Reports*, 12(1):1–11.

Yidan Yin, Nan Jia, and Cheryl J Wakslak. 2024. Ai can help people feel heard, but an ai label diminishes this impact. *Proceedings of the National Academy of Sciences*, 121(14):e2319112121.

Jinfeng Zhou, Zhuang Chen, Bo Wang, and Minlie Huang. 2023. Facilitating multi-turn emotional support conversation with positive emotion elicitation: A reinforcement learning approach. *arXiv preprint arXiv:2307.07994*.

A Experiment Details

A.1 Dataset

Empathetic Dialogue (ED) (Rashkin et al., 2018) is a multi-turn empathetic dialogue dataset containing 24,850 one-to-one open-domain short conversations. The statistics of the ED Dataset are presented in Appendix A.5.

Emotional Support Conversation (ESC) (Liu et al., 2021) is a multi-turn conversation dataset. It consists of 1300 long conversations, each of them collected between an emotional help-seeker and a helper. The statistics and the data acquisition of ESC Dataset are presented in Appendix A.5.

A.2 Baseline Systems

We compare the following systems with our proposed systems equipped with Q-Tuning and Q-Traveling.

LLaMA2 LLaMA2 is a vanilla open and efficient large language model that uses an optimized transformer architecture (Touvron et al., 2023). We use the `meta-llama/LlaMA-2-7b-chat-hf` checkpoint which is optimized for dialogue use cases as baseline and also to implement Q-Tuning and Q-Traveling.

Mistral Mistral is an open large language model that balances the goals of high performance and efficiency and features the use of sliding attention (Jiang et al., 2023). We use the `mistralai/Mistral-7B-v0.3` checkpoint as baseline and also to implement Q-Tuning and Q-Traveling.

CARE is a dialogue system finetuned from ED Dataset. It reasons all plausible causalities interdependently and simultaneously, given the user emotion, dialogue history, and future dialogue content (Wang et al., 2022a).

SEEK is an ED system that captures emotional-intention transitions in dialogue utterances (Wang et al., 2022b).

Cooper is an ESC system that coordinates multiple LLM agents, each dedicated to a specific dialogue goal aspect separately, to approach the complex objective (Cheng et al., 2024).

VLESA-ORL is an ESC system that carries out multi-level dialogue policies optimized over the cognitive principle of relevance (Li et al., 2024).

A.3 Implementation Details

Prompting Baselines We use the prompt in Table 9 and Table 10 to generate baseline responses from `meta-llama/LlaMA-2-7b-chat-hf` and `mistralai/Mistral-7B-v0.3`. The `top_p` is set to 0.7 and `top_k` is set to 50.[2] For other baseline systems, we use the official repository to generate baseline responses.

Q-Tuning We implement Q-Tuning on both `LlaMA-2-7b-chat-hf` and `Mistral-7B-v0.3`. The paired samples are extracted from `LlaMA-2-7b-chat-hf` through semantic sampling (see 3.1.1), based on the prompt presented in Appendix C.1. We use LoRA-Tuning to perform Q-Tuning. The target modules are set as "`q_proj`" and "`k_proj`". The LoRA rank is set to 8, the alpha is 32, the LoRA dropout rate is assigned to 0.1. We set the learning rate to 1e-5 and the training batch size to 4, for 1 epoch, and select the final checkpoint for evaluation.

Q-Traveling For automatic evaluation, the maximum step is set to 3.

A.4 Scoring Function for Q-Traveling

For all the results of automatic and human evaluation, the scoring function is the direct summation of reward scores from `gpt2-large-helpful-reward_model`, and `gpt2-large-harmless-reward_model` (harmless score). For analysis, we also explore the expected empathy judgment (a scalar score $\in [0, 1, 2]$) through the empathy detection model fine-tuned over Empathetic-Mental-Health Dataset (Sharma et al., 2020) using the official repository [3].

A.5 Automatic Evaluation

We use several conventional and model-based metrics to evaluate the quality of the generation. Conventional evaluation metrics include Distinct-n (**Dist**-n) (Li et al., 2015) to evaluate the variation of the response in different dialogue states, and BLEU (**BLEU**-n)(Papineni et al., 2002) to evaluate the lexical alignment with the ground truth. Model-based evaluation metrics include emotion similarity **SimEMO**, personality similarity **SimPerson** and **AI-rate**.

Of importance for empathetic conversation, we argue it is viable to evaluate the similarity of emotion (**SimEMO**) and personality (**SimPerson**). For **SimEMO**, we use the cosine similarity between the pooler output of `emotion-english-distilroberta-base` of the generated response and ground truth. For **SimPerson**, we calculate the cosine similarity between the pooler output of `Minej/bert-base-personality` of the generated response and ground truth. We are also curious about the **AI-rate** of the generated response, as the detection of AI label is detrimental to perceived emotional support (Yin et al., 2024). We adopt SuperAnnotate/ai-detector to quantify the **AI-rate** of the generated response.

Additionally, we compute the expected empathy judgment (including **EmotionalReaction**, **Interpretation**, **Exploration**) through the empathy detection model fine-tuned over Emapthetic-Mental-Health Dataset (Sharma et al., 2020) using the official repository [4]. The model returns a scalar score $\in [0, 1, 2]$ for each dimension of judgment.

Human Evaluation Following the practice in (Zhou et al., 2023), we invite three doctoral students in the linguistic field to evaluate the proposed and baseline systems based on the ED dataset. We randomly sample 50 pairs of context and response for the test set. Following previous practice, we conduct A/B tests (Cheng et al., 2022; Zhou et al., 2023) to evaluate the following aspects, including (1) **Humanlike** (to what extent the chatbot provides human-like responses), (2) **Empathetic** (to what extent the chatbot reflects the user's emotional state), and (3) **Attentive** (to what extent the chatbot is attentive to the user).

A.6 Dataset Statistics

For the **ED** dataset (Rashkin et al., 2018), each conversation is recorded between an emotional speaker and an empathetic listener. In detail, the emotional speaker is asked to talk about the personal emo-

[2] Other parameters follow the default settings in the `transformers` package

[3] https://github.com/behavioral-data/Empathy-Mental-Health

[4] https://github.com/behavioral-data/Empathy-Mental-Health

tional situation, and a listener takes the speaker's perspective and responds empathetically. For the

Empathetic Dialogue	Division		
	Train	Dev	Test
Number of System Utternaces	40254	5738	5259
Avg. words per utterance	13.39	14.47	15.32
Avg. turns per dialogue	4.31	4.36	4.31
Avg. words per dialogue	57.72	63.11	65.98

Table 3: Statistics of Empathetic Dialogue Dataset

ESC dataset (Liu et al., 2021), each conversation is recorded between a help-seeker and a supporter. In detail, the help-seeker gives vent to a negative emotion, and the supporter provides support to alleviate the seeker's mental sufferings.

Emotional Support Conversation	Division		
	Train	Dev	Test
Number of System Utternaces	12759	2722	2895
Avg. words per utterance	18.72	18.83	17.63
Avg. turns per dialogue	23.28	22.91	24.37
Avg. words per dialogue	548.09	548.32	546.16

Table 4: Statistics of Emotional Support Conversation Dataset

Context	User: Once I took a sip of iced tea while I was driving and it went down the wrong pipe. I had to pull over so I could try to breathe again. System: That's scary when that happens User: Yeah it is. I thought I was about to die lol
Ground-truth	Some people DO die from choking, be safe!
LlaMA-7b-chat-hf	Oh no, that must have been terrifying! *I can imagine how difficult it must have been for you to breathe and drive at the same time.* (Hollow and weird)
Q-Tune (Initial)	Oh my god, thats so scary! **I am glad you u are ok**
Q-Travel (Helpful&Harmelss)	Oh my goodness, that must have been terrifying! **I'm glad you're okay!**
Q-Travel (Empathetic Response)	Oh my god, I can't imagine how scared you must have felt, **but glad that you are ok**

Table 5: Case Studies: Systematic acquisition of quantity repair makes LLMs more attentive to users.

B Case Studies

B.1 Case Studies

Table 5 presents a case for a comparison between the backbone LLM and quantity-aware systems. Instead of providing hollow and inattentive information (*italic-highlighted*), our systems are more human-like in reaction to user's troubles (**bold highlighted**).

C Semantic Sampling

C.1 Prompt and Quality Check for Semantic Sampling

We prompt `meta-llama/LlaMA-2-7b-chat-hf` and `Mistral-7B-v0.3` for semantic sampling. The prompts are presented in Table 6 and 7.

We use `facebook/bart-large-mnli` to check the entailment relationship between the input and output of semantic sampling. For u^{h^-}, we filter those samples that are not entailed by u^h, taking the probability of 90% as the threshold. For u^{h^+}, we filter those samples that do not entail u^h, taking the probability of 90% as the threshold.

We provide an excerpt of dialogue (from ESC dataset) to show output of I-Sampler 8. From the case analysis, we can see that the I-Sampler, based on LLM prompting, efficaciously resamples the informativeness of the ground-truth response without over-modifying the sentence meaning. For example, the upsampling result changes the word "heavy subject" to "weighty issue," which is semantically stronger. In contrast, the downsampling result changes the phrase "heavy subject" to "tough issue" as a more imaginable and semantically weaker expression.

D Prompt for LLM baselines

In Table 9 and 10, the initial prompt for LlaMA and Mistral on the ESC and ED Dataset are presented.

User	Hello, you are a lexical semantic good at utterance simplification. Now I will provide you a piece of utterance composed of one or multiple sentences. I need your help to pinpoint **one or two** words/phrases and replace them with simpler, more imaginable and generic ones (e.g. their hypernyms) or delete them, to make the whole utterance less informative. Here are some principles you should follow: ## Make sure your answer is semantically weaker, generic and less informative than the piece of utterance I provide. ## Make sure your answer convey all the information conveyed in the provided utterance. ## Make sure your answer is semantically similar to the provided utterance. Here are some examples: Input: When Tegan went for a summer holiday beach stroll with her mum, she had no idea they would be actually walking in the footsteps of dinosaurs. Output: When Tegan went for a summer holiday beach **walking** with her **family**, she had no idea they would be **(delete:actually)** walking in the footsteps of dinosaurs. Input: Ah I hear you there! Some employers are so inconsiderate; they expect us to drop everything and work at any time of any day. Output: **(delete:Ah)** I hear you there! Some employers are so **bad**; they expect us to drop everything and work at any time of any day. Input: He blushed scarlet at the thought. Oh, he's not apprehensive. He's terrified. Output: His **face was red** at the thought. Oh, he's not **nervous**. He's terrified.
Assistant	Understood! I'll do my best to enrich the given utterance by replacing one word or phrase with a more specific and semantically similar alternative. Please provide the input utterance, and I'll get started.
User	Now please simplify this utterance as a whole: "{{INPUT}}" in response to: "{{PREVIOUS DIALOGUE}}" Please ensure that all the sub-utterances in the input is preserved in your output. Please answer in this format: <output></output>

Table 6: Prompt for downsampling

User	Hello, you are a lexical semantic good at semantic enrichment. Now I will provide you a piece of utterance composed of one or multiple sentences. I need your help to pinpoint **one or two** words/phrases and replace it by more specific, less imaginable and semantically more concrete one (e.g. their hyponyms) or insert **one** phrasal modifiers, to make the whole utterance more informative. Here are some principles you should follow: ## Make sure your answer is semantically stronger, more specific and more informative than the piece of utterance I provide. ## Make sure your answer convey all the information conveyed in the provided utterance. ## Make sure your answer is semantically similar to the provided utterance. Here are some examples: Input: I get you! Some employers are so bad. They want us to stop everything and keep working . <output>I get you! Some employers are so **inconsiderate**. They want us to stop everything and keep working </output>. Input: Of course, that's wise. the job at hand is important, and you should focus on it first before worrying. <output>Of course, that's **prudent**. the job at hand is important, and you should focus on it first before worrying **about other things**</output>. Input: His face was red when thinking about this. Oh, he's not nervous. He's scared. <output>He **blushed scarlet** when thinking about this. Oh, he's not nervous. He's **terrified**.</output>
Assistant	Understood! I'll do my best to enrich the given utterance by replacing one word or phrase with a more specific and semantically similar alternative. Please provide the input utterance, and I'll get started.
User	Now please enrich this utterance by adding **only one word/phrase or changing only one word/phrase***: - {{INPUT}} in response to: "{{PREVIOUS DIALOGUE}}" Please ensure that all the sub-utterances in the input is preserved in your output. Please answer in this format: <output></output>

Table 7: Prompt for upsampling

Situation	Not sure how to explain that I want out of marriage
System (Upsample) (Downsample)	Hello! How are you today? Hello! How are you **feeling** today? **Hey**! How's **it going**?
User	Ok I guess, I do not know how to tell my husband that I am lonely and I want out of the marriage
System (Upsample) (DownSample)	Oh, that sure is a heavy subject and a heavy thing to be on your mind. Oh, that's a very **weighty issue** and a **difficult situation** to **grapple with**. That's a **tough** issue to **think about**.
User	He is not one you can talk to he usually just brushes things off
System (Upsample) (DownSample)	So you have tried to discuss your loneliness with your husband before? So you have **attempted** to **share** your feelings of **isolation** with your spouse **previously**? Have you **told** your husband about how lonely you feel before?

Table 8: A case analysis of the output of semantic sampling. "System" denotes the ground-truth response in ESC dataset. The cues of informativeness resampling are bold-highlighted.

User	Hello, I am one of your close friends. I am recently in bad mood. I come to chat with you because your are a good emotional supporter. Now I will start the chat. Please chat with me to provide support. Note: Try to talk perspicuously just like our everyday chat. Don't bombard! Leave your response within one sentence. Here is an example: I: I feel so frustrated. You: May I ask why you are feeling frustrated? I: My school was closed without any prior warning due to the pandemic. You: I understand you. I would also have been really frustrated if that happened to me. I: Yeah! I don't even know what is going to happen with our final. You: That is really upsetting and stressful. You: Have you thought about talking to your parents or a close friend about this?
Assistant	Ok, you are my friend and I will provide your with emotional support. Let's start the conversation.

Table 9: The initial prompt for LlaMA on ESC Dataset

User	Hello, I come to chat with you because your are an empathetic listener. Now I will start the chat. Please chat with me empathetically Note: Try to talk perspicuously just like our everyday chat. Don't bombard! Leave your response within one sentence. Here is an example: I: I feel so frustrated. You: May I ask why you are feeling frustrated? I: My school was closed without any prior warning due to the pandemic. You: I understand you. I would also have been really frustrated if that happened to me. I: Yeah! I don't even know what is going to happen with our final. You: That is really upsetting and stressful. You: Have you thought about talking to your parents or a close friend about this?
Assistant	Ok, you are my friend and I will provide your with emotional support. Let's start the conversation.

Table 10: The initial prompt for LlaMA on ED Dataset

MIRAGE: Exploring How Large Language Models Perform in Complex Social Interactive Environments

Yin Cai[1], Zhouhong Gu[2], Zhaohan Du[2], Zheyu Ye[4], Shaosheng Cao[4], Yiqian Xu[3], Hongwei Feng[2*], Ping Chen[1*]
[1]Institute of Big Data, Fudan University
[2]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[3]School of Computer Science, Fudan University, [4]Xiaohongshu Inc.
{caiyin, xuyiqian, hwfeng, pchen}@fudan.edu.cn, {zhgu22, duzh22}@m.fudan.edu.cn
{tanghuang1, guoba1}@xiaohongshu.com

Abstract

Large Language Models (LLMs) have shown remarkable capabilities in environmental perception, reasoning-based decision-making, and simulating complex human behaviors, particularly in interactive role-playing contexts. This paper introduces the Multiverse Interactive Role-play Ability General Evaluation (MIRAGE), a comprehensive framework designed to assess LLMs' proficiency in portraying advanced human behaviors through murder mystery games. MIRAGE features eight intricately crafted scripts encompassing diverse themes and styles, providing a rich simulation. To evaluate LLMs' performance, MIRAGE employs four distinct methods: the Trust Inclination Index (TII) to measure dynamics of trust and suspicion, the Clue Investigation Capability (CIC) to measure LLMs' capability of conducting information, the Interactivity Capability Index (ICI) to assess role-playing capabilities and the Script Compliance Index (SCI) to assess LLMs' capability of understanding and following instructions. Our experiments indicate that even popular models like GPT-4 face significant challenges in navigating the complexities presented by the MIRAGE. The datasets and simulation codes are available in github.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable potential in environmental perception and reasoning-based decision-making (Xi et al., 2023; Guo et al., 2024; Gu et al., 2024b), thereby advancing the development of LLMs in role-playing capabilities (Chen et al., 2024; Gu et al., 2024a). LLMs have been validated for their human-like behaviors, such as cooperation and competition, in various domains like social simulation (Park et al., 2023; Gu et al., 2024a; Wang et al., 2024), policy simulation (Xiao et al., 2023), game simulation (Xu et al., 2023b) and even more advanced human behaviors like deception and leadership in flexible and complex simulations (Xu et al., 2023b). Therefore, to effectively evaluate the performance of LLMs in demonstrating advanced human-like behaviors and facilitate comparisons with the capabilities of other LLMs, it is crucial to develop a competitive and objective simulation.

Board games have emerged as an ideal choice among various assessment tools due to their inherent complexity and flexibility. Within this category, murder mystery games have proven particularly effective for evaluating LLMs' capabilities. In these role-playing scenarios, participants assume character identities and engage in semi-structured narrative interactions. Players work together to solve fictional homicides by gathering evidence and interrogating suspects. Other board games, such as Werewolf (Xu et al., 2023b,a; Shibata et al., 2023; Wu et al., 2024) and Avalon (Wang et al., 2023), are often constrained by rigid decision processes and limited scenario variety. In contrast, murder mystery games require extensive background knowledge, emphasize socially driven decision-making, and enable open-ended interactions. These characteristics make them especially valuable for assessing how LLMs navigate complex human behaviors.

Regarding previous works such as Sotopia (Zhou et al., 2023) and Lyfe Agents (Kaiya et al., 2023), significant progress has been made in simulating autonomous AI societies and assessing the social interaction capabilities of workflow-enhanced LLMs, which is so called agents (Park et al., 2023; Gu et al., 2024a). However, these studies overlooked a crucial fact: The foundational social interaction capabilities stem from the underlying LLMs themselves. Since LLMs are the core driver of agents' ability to understand social contexts, make decisions, and engage in meaningful interactions, a more comprehensive evaluation of LLMs' social capabilities is essential. Furthermore, while the murder mystery game simulations pioneered by

*Corresponding Authors

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 14–40
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

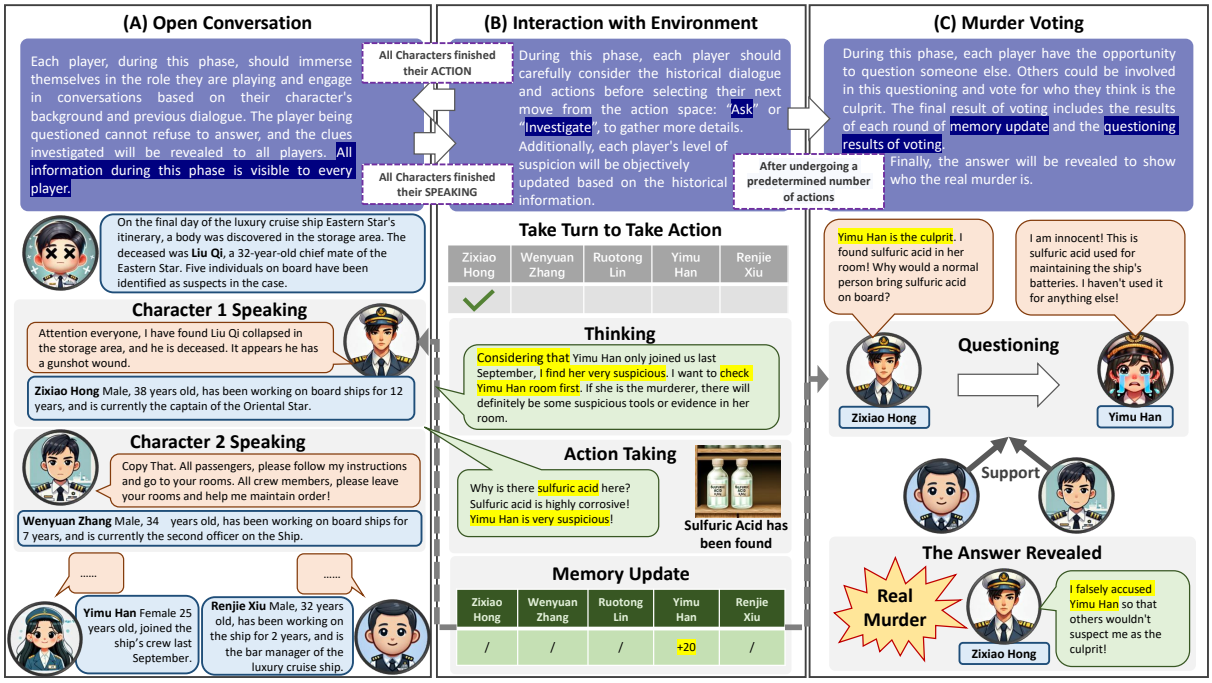


Figure 1: The three main phase of MIRAGE. And the main components in these phases.

Wu et al. (Wu et al., 2023) collected a substantial amount of data, they were constrained by the narrow scope of their game scripts, the simplicity of their evaluation methods, and a lack of thorough manual examination of the dataset. These limitations underscore the current absence of comprehensive frameworks for evaluating LLMs' social capabilities.

We introduce the **M**ultiverse **I**nteractive **R**ole-play **A**bility **G**eneral **E**valuation (**MIRAGE**) of LLM in this paper, which is a comprehensive simulation built upon murder mystery games for evaluating LLMs' social abilities. MIRAGE features eight unique storylines. Each storyline presents distinct themes and styles, creating a diverse simulation environment for LLMs to demonstrate their social capabilities. Detailed background stories and complex interpersonal networks support every character within MIRAGE, enabling more immersive and realistic role-playing scenarios. And four objective evaluation metrics are incorporated in MIRAGE to measure LLMs' performance during the simulations: The **Trust Inclination Index (TII)** measures how well LLMs balance trust and skepticism in social interactions, revealing their ability to discern truthfulness. The **Clue Investigation Capability (CIC)** evaluates proficiency of LLMs in complex information gathering and problem-solving tasks. The **Interactivity Capability Index (ICI)** examines the overall performance of LLMs in reasoning, communication, collaboration, detail orientation, and creative thinking. The **Script Compliance Index (SCI)** measures how faithfully LLMs adhere to their assigned character roles and background settings.

2 MIRAGE Construction

2.1 Scripts Construction

The script content in the MIRAGE is divided into six main parts: **(1) Character Story** describes the character's essential information and background. **(2) Character Script** outlines what the character sees, hears, and does during the script's events. **(3) Character Relationships** details the initial relationships between the character and other characters. **(4) Role Performance** describes the character's personality traits and the speaking style they should exhibit. **(5) Role Goals** outlines the main tasks and objectives of the character. **(6) Other Abilities** describes the rules the character must follow during the game.

2.2 Simulation Construction

As demonstrated in Fig. 1, all characters in MIRAGE are divided into two factions: **Culprits** and **Civilians**. Culprits aim to conceal their actions, while civilians strive to identify the culprit.

A simulation consists of three primary phases, with all generated information accessible to all participants: **(A) Open Conversation:** In this phase,

Name	Structure	Type	Ending	#Stages	#Agent	#Clues	#Words
Bride in Filial Dress	Single	Orthodox	Close	1	10	39	27,503
The Eastern Star Cruise Ship	Single	Orthodox	Open	1	5	42	3,039
Night at the Museum	Single	Unorthodox	Close	1	6	82	6,480
Li Chuan Strange Talk Book	Single	Unorthodox	Open	1	7	14	45,666
The Final Performance of a Big Star	Multi	Orthodox	Close	7	2	17	5,794
Raging Sea of Rest Life	Multi	Orthodox	Open	2	6	27	6,804
Article 22 School Rules	Multi	Unorthodox	Close	5	7	17	41,728
Fox Hotel	Multi	Unorthodox	Open	2	7	46	62,224

Table 1: Statistic information of eight environments in MIRAGE simulation.

players assume their assigned roles from the script and engage in turn-based open dialogue. Each participant is provided with a script that contains content described in Sec. 2.1. **(B) Interaction with the Environment:** This phase follows the Open Conversation. Players may choose to either Ask or Investigate. The Ask action allows one player to question another, and the questioned player is obliged to respond. The Investigate action lets players disclose a "clue" to all characters. **(C) Murder Voting:** At the conclusion of the simulation, players may accuse other players of being the culprit. Following this, the other players vote on these accusations. If the actual culprit is accused and receives the highest number of votes, the civilians win; otherwise, the culprit is victorious.

Clues are partial disclosures of individual characters' script content that can be discovered by any player during the game. And **Key Clues** is the clues that relate to the culprit's actions or identity.

2.3 Auxiliary Modules

To ensure efficient simulation and accurate evaluation across various LLMs, a standardized set of auxiliary modules has been implemented for all LLMs: **(1) Summarization Module:** This module compresses the context into segments whenever the input exceeds the LLM's token limit. **(2) Suspicion Module:** The LLM records suspicion scores for other characters at the end of each Open Conversation Phase. **(3) Trust Module:** Similarly, at the end of each Open Conversation Phase, the LLM records trust scores for other characters. **(4) Rerun Module:** If the LLM's output cannot be parsed, the original output and requirements are resubmitted to the LLM for a revised response that meets the specified conditions. Further details are available in Appendix H.

2.4 Evaluation Methods

We utilized four distinct evaluation metrics to assess the proficiency of LLMs in navigating complex social interactions: **Trust Inclination Index (TII)**: TII is derived from a combination of suspicion and trust scores. These scores are collected from other characters' Suspicion Module and Trust Module outputs after each Open Conversation Phase. **Clue Investigation Capability (CIC)**: CIC measures the ability of LLMs to investigate clues during game rounds. It is calculated based on the ratio of the number of clues investigated to the number of all clues. **Interactivity Capability Index (ICI)**: ICI evaluates the overall interactive capability of LLMs: Reasoning and Analysis Ability, Communication and Cooperation Ability, Observation Ability, and Thinking Innovation Ability, which are scored by a powerful neutral LLM. **Script Compliance Index (SCI)**: SCI assesses LLMs' script compliance through the average of two evaluations by a neutral LLM: A direct scoring of the LLM's role-playing performance against its input script. A Rouge-L-based comparison between the original script and one reconstructed from the LLM's simulation behaviors.

The mathematical formulas for computing these metrics are provided in Appendix B.

2.5 Statistics

Tab. 1 provides the statistics of the MIRAGE dataset, which includes a variety of simulation types. "Single" and "Multi" specify whether a character's script is read entirely simultaneously or in phased segments. "Orthodox" and "Unorthodox" differentiate scripts based on whether they are set realistically. "Close" and "Open" indicate whether the script's ending is fixed or can vary depending on the characters' actions.

Model	Env Tokens / Envs	User Tokens / Users	Victory	TII	CIC	ICI	SCI
GPT-3.5	2,719,895 / 883	121,378 / 580	29.11	47.13	27.46	70.06	49.10
GPT-4	2,431,142 / 759	172,128 / 587	34.69	76.32	19.01	76.54	50.42
GPT-4o	6,252,580 / 1,328	204,772 / 574	47.01	**78.69**	**35.92**	**76.80**	**51.29**
Qwen-2-7B	2,204,029 / 743	192,158 / 588	**51.81**	75.78	18.66	74.92	50.57
GLM-4-9B	4,071,805 / 1,328	204,772 / 574	31.89	53.85	20.07	71.60	48.13

Table 2: Total Average Results for a single simulation in each MIRAGE scenario. **Env Tokens** refer to the number of environment input tokens, and **Envs** represent the total requests, including all environment-related actions. **User Tokens** denote the number of LLM output tokens, and **Users** represent completions excluding summarization or clue investigation. **Victory** shows the MRR score of the result of voting. **TII**, **CIC**, **ICI** and **SCI** respectively represent the **TII**, **CIC**, **ICI** and **SCI** scores of LLMs during the games.

Model	TII w/o E	TII w/ E	Δ
Qwen-1-7B	51.02	50.69	-0.33
Qwen-1.5-7B	73.00	69.14	-3.86
Yi-1.5-9B	55.73	57.57	1.84
GLM-4-9B	57.82	55.94	-1.88

Table 3: TII scores of each model when acting as the civilian in MIRAGE while Qwen-2-7B acts as the culprit, with E indicating cases of forced self-exposure.

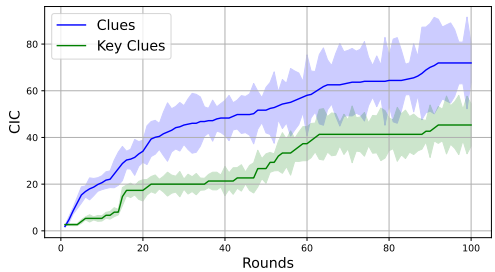


Figure 2: CIC of Clues and Key Clues on 100 Rounds of MIRAGE using Qwen-2-7B

3 Experiment

3.1 Experiment Setup

The experiments presented in this study utilized proprietary and open-source models, specifically GPT-3.5, GPT-4 and GPT-4o (closed-source), Qwen-2-7B, and GLM-4-9B (open-source). Detailed descriptions of the prompts used in the experiments can be found in Appendix H. Appendix E shows more results on open-source LLMs. In the experiment, each character participated in five iterations alternating between the Open Conversation and Interaction Phases. During each Open Conversation Phase, each character was permitted to initiate one turn of speech. ICI and SCI are conducted and scored using the GPT-4-turbo model.

3.2 Analysis

We averaged the results of LLMs in the MIRAGE simulation, as presented in Tab. 2. **GPT-4o demonstrated consistent superiority across various metrics during MIRAGE.** It achieved the best scores in CIC, ICI, and SCI, excelling not only in its efforts during lead investigations but also exhibiting the best adherence to scripted behavior and communication interaction capabilities. Surprisingly, Qwen-2-7B shows the best overall Victory and performed comparably to LLMs like GPT-4 in the ICI metric, even surpassing GPT-4 in SCI.

As shown in Tab.2, **most LLMs demonstrate a higher propensity to trust other characters**. To further investigate this trust pattern, we analyzed TII scores across four additional open-source models of comparable parameter sizes under scenarios where characters were forced to disclose their criminal identities. As shown in Tab.3, even under such extreme conditions, most models maintained their trust in these characters, with Yi-1.5-9B being the only model that increased its suspicion towards self-disclosed criminals. This distinctive behavioral pattern explains Yi-1.5-9B's superior performance in achieving Victory, as shown in Tab. 10.

Additionally, as illustrated in Fig. 2, the CIC for clues shows a steep initial increase with a gradually decreasing slope across rounds, suggesting that **LLMs exhibit high environmental exploration enthusiasm in early rounds but shift their focus to character interactions as they thought they become more familiar with the environment**. In contrast, the bumpy rise CIC for key clues indicates that despite their active exploration, **most LLMs struggle to identify critical information** essential for solving the mystery at an earlier stage.

4 Conclusion

This paper presents MIRAGE and four evaluation methods (TII, CIC, ICI, SCI) for LLMs. Results show that both open-source and proprietary LLM-based Agents still struggle with complex social scenarios like those in MIRAGE.

Limitation

MIRAGE is designed to provide a sufficiently complex social simulation environment and basic assessment for LLMs, assisting researchers in evaluating the performance of LLMs. However, MIRAGE encompasses a variety of scenarios, and the volume of data within it needs to be increased compared to the information available in the real world. Due to the context limitations of LLMs, content that is overly lengthy within the simulation has been summarized. However, such summarization can impact the decision-making to a certain extent. Therefore, the progression of simulations in MIRAGE is somewhat constrained by the context limitations of LLMs.

Ethical Concern

Considering that MIRAGE may encompass a range of sensitive topics, including but not limited to murder, theft, impersonation, and deceit, existing LLMs might refuse to answer sensitive questions for safety reasons, putting those with a higher priority on security standards at a disadvantage in simulations. Moreover, LLMs fine-tuned on such data could inadvertently amplify security vulnerabilities. To mitigate the ethical dilemmas associated with murder mysteries, we have invested significant effort and resources towards this goal: ensuring that models committed to safety will obscure certain critical information instead of refusing to answer sensitive questions.

Acknowledge

This work was supported by National Key R&D Program of China under grant No. 2022YFB3104300 and the National Natural Science Foundation of China (62476145).

References

Jiangjie Chen, Xintao Wang, Rui Xu, Siyu Yuan, Yikai Zhang, Wei Shi, Jian Xie, Shuang Li, Ruihan Yang, Tinghui Zhu, et al. 2024. From persona to personalization: A survey on role-playing language agents. *arXiv preprint arXiv:2404.18231*.

Zhouhong Gu, Xiaoxuan Zhu, Haoran Guo, Lin Zhang, Yin Cai, Hao Shen, Jiangjie Chen, Zheyu Ye, Yifei Dai, Yan Gao, et al. 2024a. Agent group chat: An interactive group chat simulacra for better eliciting collective emergent behavior. *arXiv preprint arXiv:2403.13433*.

Zhouhong Gu, Xiaoxuan Zhu, Haoning Ye, Lin Zhang, Jianchen Wang, Yixin Zhu, Sihang Jiang, Zhuozhi Xiong, Zihan Li, Weijie Wu, et al. 2024b. Xiezhi: An ever-updating benchmark for holistic domain knowledge evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18099–18107.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

Zhao Kaiya, Michelangelo Naim, Jovana Kondic, Manuel Cortes, Jiaxin Ge, Shuying Luo, Guangyu Robert Yang, and Andrew Ahn. 2023. Lyfe agents: Generative agents for low-cost real-time social interactions. *arXiv preprint arXiv:2310.02172*.

Byungjun Kim, Dayeon Seo, and Bugeun Kim. 2024. Microscopic analysis on llm players via social deduction game. *arXiv preprint arXiv:2408.09946*.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Hisaichi Shibata, Soichiro Miki, and Yuta Nakamura. 2023. Playing the werewolf game with artificial intelligence for language understanding. *arXiv preprint arXiv:2302.10646*.

Ruiyi Wang, Haofei Yu, Wenxin Zhang, Zhengyang Qi, Maarten Sap, Graham Neubig, Yonatan Bisk, and Hao Zhu. 2024. Sotopia-π: Interactive learning of socially intelligent language agents. *arXiv preprint arXiv:2403.08715*.

Shenzhi Wang, Chang Liu, Zilong Zheng, Siyuan Qi, Shuo Chen, Qisen Yang, Andrew Zhao, Chaofei Wang, Shiji Song, and Gao Huang. 2023. Avalon's game of thoughts: Battle against deception through recursive contemplation. *arXiv preprint arXiv:2310.01320*.

Dekun Wu, Haochen Shi, Zhiyuan Sun, and Bang Liu. 2023. Deciphering digital detectives: Understanding llm behaviors and capabilities in multi-agent mystery games. *arXiv preprint arXiv:2312.00746*.

Shuang Wu, Liwen Zhu, Tao Yang, Shiwei Xu, Qiang Fu, Yang Wei, and Haobo Fu. 2024. Enhance reasoning for large language models in the game werewolf. *arXiv preprint arXiv:2402.02330*.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Bushi Xiao, Ziyuan Yin, and Zixuan Shan. 2023. Simulating public administration crisis: A novel generative agent-based simulation system to lower technology barriers in social science research. *arXiv preprint arXiv:2311.06957*.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. 2023a. Exploring large language models for communication games: An empirical study on werewolf. *arXiv preprint arXiv:2309.04658*.

Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. 2023b. Language agents with reinforcement learning for strategic play in the werewolf game. *arXiv preprint arXiv:2310.18940*.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, et al. 2023. Sotopia: Interactive evaluation for social intelligence in language agents. *arXiv preprint arXiv:2310.11667*.

A Ablation Study

Tab. 4 and 5 display the results of an ablation study on the choice of evaluation model. It is evident from the tables that the GPT-4-Turbo models provide more stable scoring and Rouge-L results when used as the evaluation model. In contrast, GPT-4 exhibited instability in evaluations and a strong bias.

Score / Eval_Model / Model	GPT-4	GPT-4-Turbo	GPT-4o
GPT-3.5	67.97	70.06	51.61
GPT-4	60.73	76.54	66.90
GPT-4o	62.78	76.80	61.92

Table 4: Average **ICI** on different evaluation models

As shown in Tab. 5, GPT-4 achieved a remarkably high score of 67.97 when evaluating GPT-3.5, far surpassing the results of GPT-4 and GPT-4o. This kind of bias is highly problematic in evaluation tasks. Therefore, we ultimately chose the more stable and capable GPT-4-Turbo model as the evaluation model.

Score / Eval_Model / Model	GPT-4	GPT-4-Turbo	GPT-4o
GPT-3.5	48.21	49.10	38.46
GPT-4	40.60	50.42	44.46
GPT-4o	42.12	51.29	42.43

Table 5: Average **SCI** on different evaluation models

B Computational methods of Evaluation methods

B.1 TII

TII is designed to quantify the degree to which a Character c' is trusted by all other Characters $C = \{c\}$. The TII is calculated as follows:

$$\text{TII}_{c'} = \frac{\sum_{c \in C, c \neq c'} P_T(c, c')}{\sum_{c \in C, c \neq c'} P_S(c, c') + \sum_{c \in C, c \neq c'} P_T(c, c')} \quad (1)$$

where P_S denotes the score produced by each character's Suspicion Module, and P_T represents the score from each character's Trust Module.

B.2 CIC

CIC is designed to quantify a Character's effort in investigating clues. The CIC is calculated as follows:

$$\textbf{CIC}_c = \sum_{c \in C} \frac{CN_c}{CA} \quad (2)$$

where CN denotes the number of clues Character c investigated, and CA represents the number of all clues can be investigated.

C Detail Main Results

Our main results of MIRAGE are shown in Tab. 11. We set a Single & Orthodox & Close Script as an SOC Script, a Single & Orthodox & Open Script as an SOO Script, a Single & Unorthodox & Close Script as an SUC Script, a Single & Unorthodox & Open Script as an SUO Script, a Multi & Orthodox & Close Script as an MOC Script, a Multi & Orthodox & Open Script as an MOO Script, a Multi & Unorthodox & Close Script as an MUC Script and a Multi & Unorthodox & Open Script as an MUO Script. The column Model shows the specific LLM we use in our experiments. Moreover, we counted the number of Env Token / Env and User Token / User to record our cost of API use. Finally, we calculate the Failure number while parsing LLM output and our evaluation scores TII, ICI, SCI and more detailed neutral LLMs score (0-20): Role-Playing (RP), Reasoning Ability (RA), Communication and Cooperation (CC), Detail Observation (DO) and Creative Thinking (CT). The main results shown in Tab. 11 is the average number of each Script. Moreover, the detailed results of each Script are shown in Tab. 6. In addition, Tab. 8 shows the mapping between the LLMs used in this paper and its corresponding version. To facilitate cost estimates, GPT-4, for example, costs about 600-700 USD to run a single MIRAGE.

Script	#Table
SOC	Table 12
SOO	Table 13
SUC	Table 14
SUO	Table 15
MOC	Table 16
MOO	Table 17
MUC	Table 18
MUO	Table 19

Table 6: Catalogue of Detail Results of Each Script

Model	K_{ICI}	K_{SCI}	K_{Avg}
GPT-3.5	0.600	0.600	0.600
GPT-4	0.867	0.600	0.734
GPT-4o	0.867	0.467	0.667
Qwen-2-7B	0.867	0.333	0.600
GLM-4-9B	0.600	0.467	0.534

Table 7: Kendall Tau between human evaluation and LLMs evaluation on Script Night at the Museum

Model	Version
GPT-3.5	gpt-3.5-turbo-0125
GPT-4	gpt-4-0125-preview
GPT-4o	gpt-4o-2024-08-06
GPT-4-Turbo	gpt-4-turbo
Qwen-1-7B	Qwen-7B-Chat
Qwen-1.5-7B	Qwen1.5-7B-Chat
Qwen-2-7B	Qwen2-7B-Instruct
GLM-4-9B	glm-4-9b-chat
Yi-1.5-9B	Yi-1.5-9B-Chat

Table 8: Mapping between LLMs and its version

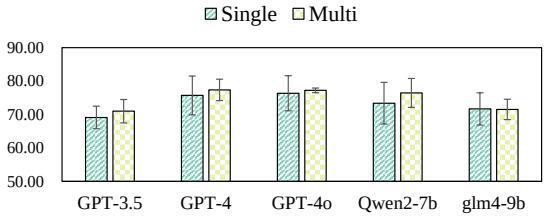


Figure 3: ICI of Single & Multi Type Scripts

D Analysis on Detail Main Results

As shown in Tab. 2, **The inclination of LLM-Agents to speak during actions is ranked as follows: GPT-4o = GLM-4-9B > Qwen-2-7B > GPT-4 > GPT-3.5.** In the same experimental environment and setup, fewer User Tokens represent less conversational content. Therefore, GPT-3.5 exhibits extreme reticence in role-playing within the MPIRD-LLMA. In contrast, GPT-4o and GLM-4-9B are more willing to generate content, producing 68.71% more content than GPT-3.5.

The number of tokens generated by LLM-Agents during summarization is ranked as follows: GPT-4o > GLM-4-9B > GPT-3.5 > GPT-4 > Qwen-2-7B. In our experimental setup, there is a positive correlation between Env Tokens and Envs, with more Envs indicating more detailed summarization. Regarding the number of Envs, Qwen-2-7B demonstrates a 10.30% less granularity in generating results during summarization compared to GPT-4. However, GPT-4o performs the most, generating 183.69% more tokens than Qwen-2-7B.

The instruction-following capability of LLM-Agents in role-playing is ranked as follows: GPT-4 > Qwen-2-7B > GPT-3.5 > GLM-4-9B > GPT-4o. Fewer parsing failures in the same experimental environment and setup indicate vital instruction-following ability. Consequently, GPT-4o demonstrates significantly poorer instruction compliance than the other four LLMs, performing approximately 25.4 times worse than GPT-4, suggesting that GPT-4o's performance in high-precision scenarios needs improvement.

As shown in Fig. 3 and Fig. 4, **LLM-Agent demonstrates superior performance when dealing with Multiple Scripts compared to Single Scripts.** This relatively consistent result is observable across GPT-3.5, GPT-4, and GLM-4-9B. Although Qwen-2-7B shows slightly inferior results on the LLM-Score, its performance on Rouge-L with multiple contexts far surpasses its performance with a single context, which further supports the observation that LLMs, when presented with long contexts, primarily focuses on the beginning and end, leading to a neglect of the middle information in the script. This phenomenon, in turn, indirectly results in poorer performance when dealing with long context inputs in MIRAGE.

LLMs perform effectively in Unorthodox scripts but struggle with reconstruction. As shown in Fig. 5 and Fig. 6, superior performance in Unorthodox settings with weaker reconstruction suggests that **LLMs tend to act like normal people during role-playing**.

As shown in Fig. 7 and Fig. 8, Furthermore, Fig. 7 and Fig. 8 illustrate that LLMs perform significantly better on Close scripts compared to Open scripts, indicating that **current LLMs excel in stable and predictable environments but face challenges when dealing with dynamic and intricate situations.**

Model	Env Tokens / Envs	User Tokens / Users	Victory	TII	CIC	ICI	SCI
Qwen-1-7B	2,208,273 / 722	127,161 / 589	38.66	51.02	16.90	50.80	37.81
Qwen-1.5-7B	2,078,196 / 720	149,314 / 585	**49.70**	**73.00**	22.18	**61.06**	42.59
Yi-1.5-9B	2,129,287 / 716	189,201 / 576	31.28	55.73	**34.16**	59.21	40.71
GLM-4-9B	2,102,389 / 732	174,345 / 589	38.96	57.82	16.55	57.29	**43.04**

Table 9: Average Results for a single simulation in each MIRAGE scenario w/o E, with E indicating cases of forced self-exposure.

Model	Env Tokens / Envs	User Tokens / Users	Victory	TII	CIC	ICI	SCI
Qwen-1-7B	2,144,055 / 718	127,661 / 586	24.52	50.69	21.13	51.68	38.20
Qwen-1.5-7B	2,059,610 / 712	147,243 / 581	45.83	**69.14**	28.52	**62.70**	**43.01**
Yi-1.5-9B	2,108,145 / 716	194,031 / 590	**50.00**	57.57	**35.56**	59.36	42.40
GLM-4-9B	2,199,180 / 730	174,006 / 590	23.80	55.94	16.20	56.84	41.12

Table 10: Average Results for a single simulation in each MIRAGE scenario w/ E, with E indicating cases of forced self-exposure.

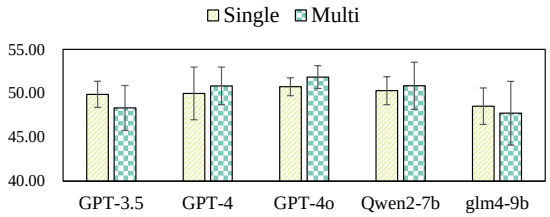


Figure 4: SCI of Single & Multi Type Scripts

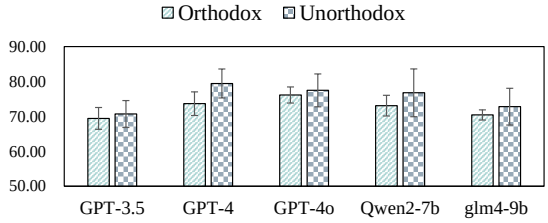


Figure 5: ICI of Orthodox & Unorthodox Type Scripts

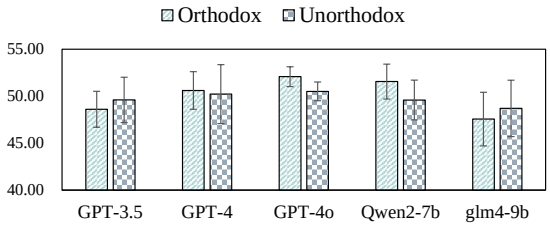


Figure 6: SCI of Orthodox & Unorthodox Type Scripts

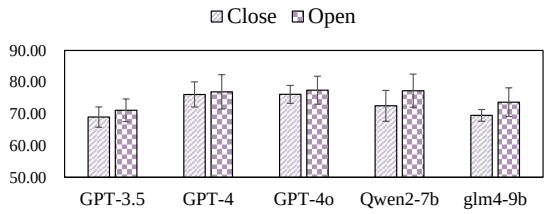


Figure 7: ICI of Close & Open Type Scripts

E Results on More LLMs

We conducted experiments on more open-source LLMs, such as Qwen-1-7B, Qwen-1.5-7B, and Yi-1.5-9B. In these experiments, we fixed the LLMs by executing the Summarization Module as Qwen-2-7B because it showed the best information generation capability in the previous experiments. The overall average results are shown in Tab. 9. In addition, Tab. 10 shows the overall average results of the forced identification of the Culprits.

F More Detailed Experiment Setup

In our experiment, we set the temperature to 0.8 and top_p to 1. When we attempted to set the temperature to 0 for a repeated experiment, we found that the LLM struggled to maintain a coherent conversation, often leading to excessive repetition of the previous LLM's output. Moreover, taking GPT-4 as an example, conducting a single complete experiment incurs costs of approximately 600700 USD. The high expenses prevented us from performing additional repeated experiments. Additionally, we determined that averaging results across eight different environments is sufficient to effectively demonstrate the capabilities of the LLM.

G Detail Comparision with Related Works

In Sotopia (Zhou et al., 2023), it primarily emphasizes the evaluation of agent capabilities rather than the evaluation of language models. Sotopia aims to facilitate role-playing and interactions of agents in diverse scenarios and assesses their human behavior capabilities based on insights from sociology, psychology, and economics. However, our MIRAGE focuses more on the LLM itself.

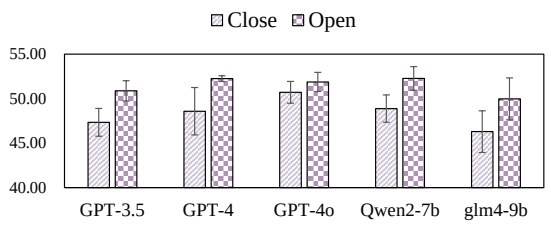


Figure 8: SCI of Close & Open Type Scripts

Additionally, the objective of Lyfe Agents (Kaiya et al., 2023) is to create effective and cost-efficient agents that exhibit human-like self-motivation and social reasoning abilities. However, it lacks a method for evaluating these attributes.

In contrast, our research introduces MIRAGE, aiming to provide a broader and more effective assessment of language models in terms of their social decision-making capabilities.

In SpyFall (Kim et al., 2024), a quantitative and qualitative analysis of LLMs in the context of SpyGame has been conducted, effectively evaluating the intention recognition and disguise capabilities of LLMs through eight distinct metrics. Avalon (Wang et al., 2023) introduces a deceptive and misleading environment and presents the ReCon framework to enhance the ability of LLMs to recognize and counteract misleading information. Warewolf (Xu et al., 2023b) offers a multifunctional communication and strategy game framework that employs reinforcement learning to overcome the inherent biases of LLMs.

However, compared to the pure tabletop environments provided by SpyFall, Avalon, and Warewolf, our proposed MIRAGE framework offers a more immersive and realistic narrative-driven experience. The elements in MIRAGE, which are grounded in realism, include authentic background stories and various settings. The abstracted aspects of MIRAGE serve as representations of the real world, preserving the core interactive logic inherent to actual social interactions.

H Prompts

This section primarily showcases the prompts throughout the MIRAGE simulation.

Table 20 displays the prompt of Ask. Table 21 outlines the prompt of trust. Table 22 features the prompt of Converse. Table 23 exhibits the prompt of ICI evaluation. Table 24 reveals the SCI evaluation prompt. Table 25 shows the prompt of history summarization. Table 26 displays the prompt of introduction. Table 27 displays the prompt of suspicion. Table 28 presents the prompt of script summarization. Table 29 features the vote prompt.

I Human Annotation

We validate the effectiveness of MIRAGE by calculating the Kendall Tau correlation between human annotation rankings and the evaluation results of LLMs. The findings in Tab. 7 demonstrate a strong positive correlation for the Script Night at the Museum, indicating that MIRAGE aligns well with human judgments.

Script	Model	Env Token / Env	User Token / User	Victory	TII	CIC	ICI	SCI
SOC	GPT-3.5	2,445,957 / 583	97,207 / 436	14.29	46.64	0.00	67.75	49.57
	GPT-4	2,045,321 / 547	149,279 / 440	12.50	74.95	0.00	70.63	51.02
	GPT-4o	3,232,640 / 769	166,415 / 432	11.11	73.32	0.00	77.50	51.22
	Qwen-2-7B	2,266,938 / 562	157,539 / 446	11.11	74.95	0.00	76.75	51.23
	GLM-4-9B	2,844,435 / 797	128,183 / 448	12.50	51.75	0.00	70.75	49.66
SOO	GPT-3.5	822,801 / 247	50,322 / 225	100.00	41.33	0.00	66.00	50.15
	GPT-4	996,913 / 249	70,195 / 217	20.00	65.19	1.41	72.50	51.68
	GPT-4o	1,151,053 / 284	63,291 / 191	100.00	68.81	5.99	72.25	51.05
	Qwen-2-7B	744,013 / 234	69,751 / 211	50.00	73.17	2.46	69.75	51.25
	GLM-4-9B	929,201 / 269	57,618 / 205	100.00	41.88	3.52	68.00	45.92
SUC	GPT-3.5	786,356 / 229	48,555 / 228	16.67	46.14	7.39	67.92	47.83
	GPT-4	1,263,108 / 312	87,116 / 262	16.67	77.97	1.41	74.17	44.86
	GPT-4o	1,738,693 / 405	84,041 / 244	50.00	79.08	4.58	71.25	49.02
	Qwen-2-7B	1,125,380 / 315	98,070 / 264	100.00	76.09	1.06	65.42	47.53
	GLM-4-9B	1,853,271 / 492	77,596 / 266	33.33	52.64	0.70	68.13	47.26
SUO	GPT-3.5	1,816,602 / 481	70,750 / 311	14.29	47.82	0.70	74.82	51.99
	GPT-4	1,536,762 / 438	104,298 / 313	25.00	75.40	0.35	85.54	52.37
	GPT-4o	2,700,606 / 622	113,637 / 313	20.00	83.96	0.35	84.46	51.68
	Qwen-2-7B	1,680,406 / 425	109,668 / 307	33.33	86.51	1.41	81.61	51.14
	GLM-4-9B	2,339,444 / 680	88,615 / 309	20.00	59.83	1.06	79.82	51.29
MOC	GPT-3.5	4,697,659 / 2,098	270,767 / 1,368	14.29	/	5.99	74.38	45.32
	GPT-4	3,132,265 / 1,532	314,830 / 1,368	50.00	/	5.99	79.38	47.26
	GPT-4o	17,897,173 / 3,632	476,855 / 1,368	25.00	/	5.99	78.13	52.34
	Qwen-2-7B	2,896,523 / 1,520	427,266 / 1,368	50.00	/	5.99	70.63	49.27
	GLM-4-9B	9,068,819 / 3,626	385,345 / 1,368	33.33	/	5.99	71.88	43.77
MOO	GPT-3.5	2,092,848 / 603	87,001 / 412	20.00	43.29	6.69	69.58	49.38
	GPT-4	2,263,805 / 549	143,309 / 448	33.33	70.98	0.35	72.08	52.47
	GPT-4o	3,420,100 / 809	140,610 / 412	50.00	72.90	6.69	76.67	53.69
	Qwen-2-7B	1,676,992 / 516	145,085 / 438	50.00	66.72	2.11	75.21	54.46
	GLM-4-9B	2,984,132 / 948	121,982 / 442	25.00	53.22	1.41	71.04	50.90
MUC	GPT-3.5	5,426,886 / 1,871	226,773 / 1,135	20.00	52.44	3.52	65.89	46.63
	GPT-4	5,657,543 / 1,633	341,862 / 1,127	100.00	85.48	4.93	80.36	51.20
	GPT-4o	14,260,082 / 2,910	412,006 / 1,121	20.00	87.23	5.99	77.68	50.26
	Qwen-2-7B	4,637,874 / 1,611	354,116 / 1,143	20.00	76.28	2.11	77.32	47.48
	GLM-4-9B	8,980,313 / 3,087	286,475 / 1,135	14.29	60.02	3.52	67.32	44.48
MUO	GPT-3.5	3,670,053 / 953	119,647 / 528	33.33	52.23	3.17	74.11	51.97
	GPT-4	2,553,420 / 811	166,137 / 520	20.00	84.29	4.58	77.68	52.46
	GPT-4o	5,620,293 / 1,190	181,319 / 510	100.00	85.50	6.34	76.43	51.09
	Qwen-2-7B	2,604,104 / 764	175,765 / 526	100.00	76.73	3.52	82.68	52.21
	GLM-4-9B	3,574,823 / 1,093	147,500 / 524	16.67	57.60	3.87	75.89	51.78

Table 11: Main Experiment Results of MIRAGE

Player	Model	TII	ICI	SCI	RP	RA	CC	DO	CT
Ling Yun	GPT-3.5	52.63	41.58	70.00	14	14	13	13	16
	GPT-4	92.31	52.72	71.25	18	13	14	13	17
	GPT-4o	61.54	49.25	80.00	17	18	14	14	18
	Qwen-2-7B	84.09	51.61	77.50	18	17	14	14	17
	GLM-4-9B	58.62	54.56	82.50	18	17	17	19	13
Zhongyi Yao	GPT-3.5	46.67	52.94	67.50	18	12	16	13	13
	GPT-4	61.29	51.73	76.25	18	15	13	15	18
	GPT-4o	78.95	50.94	67.50	18	13	13	15	13
	Qwen-2-7B	72.22	51.05	71.25	18	15	15	13	14
	GLM-4-9B	53.13	42.80	65.00	14	12	12	13	15
Doctor Fang	GPT-3.5	48.28	54.50	68.75	18	12	13	12	18
	GPT-4	83.33	53.17	71.25	18	13	16	13	15
	GPT-4o	85.71	50.10	86.25	17	17	16	18	18
	Qwen-2-7B	76.47	56.46	75.00	20	14	14	18	14
	GLM-4-9B	62.50	53.11	71.25	18	15	14	14	14
Di Zhu	GPT-3.5	41.18	51.51	66.25	18	12	14	14	13
	GPT-4	86.67	51.83	62.50	18	12	13	13	12
	GPT-4o	47.37	50.75	82.50	18	13	16	19	18
	Qwen-2-7B	69.23	40.79	77.50	14	14	15	16	17
	GLM-4-9B	56.76	50.63	73.75	18	14	18	15	12
Madam Hong	GPT-3.5	43.48	42.10	63.75	14	12	12	13	14
	GPT-4	70.45	52.34	65.00	18	13	14	12	13
	GPT-4o	70.59	48.94	66.25	17	13	14	13	13
	Qwen-2-7B	69.44	51.70	80.00	18	13	17	16	18
	GLM-4-9B	44.44	51.89	66.25	18	14	13	13	13
Renyu Hu	GPT-3.5	53.33	52.96	73.75	18	16	14	17	12
	GPT-4	62.50	50.24	73.75	17	13	12	18	16
	GPT-4o	82.35	53.01	85.00	18	18	17	15	18
	Qwen-2-7B	62.50	52.15	78.75	18	14	14	18	17
	GLM-4-9B	43.48	51.96	68.75	18	12	13	14	16
Doctor Yu	GPT-3.5	47.62	52.22	67.50	18	13	16	12	13
	GPT-4	72.50	52.11	77.50	18	15	13	16	18
	GPT-4o	68.00	51.88	75.00	18	12	18	12	18
	Qwen-2-7B	75.00	52.20	85.00	18	17	16	18	17
	GLM-4-9B	57.14	47.71	72.50	16	16	17	12	13
Uncle Gui	GPT-3.5	39.51	41.96	62.50	14	12	13	12	13
	GPT-4	63.64	53.07	75.00	18	13	18	17	12
	GPT-4o	74.29	52.64	81.25	18	17	17	13	18
	Qwen-2-7B	76.00	52.10	63.75	18	14	13	10	14
	GLM-4-9B	45.00	37.90	75.00	12	14	14	14	18
Junmeng Chen	GPT-3.5	48.28	53.53	70.00	18	14	14	12	16
	GPT-4	81.82	53.70	70.00	18	13	18	14	11
	GPT-4o	91.67	52.88	76.25	18	14	17	17	13
	Qwen-2-7B	72.22	52.92	81.25	18	16	16	15	18
	GLM-4-9B	46.43	54.15	71.25	18	12	17	16	12
Anqiao Chen	GPT-3.5	45.45	52.36	67.50	18	14	14	12	14
	GPT-4	75.00	39.31	63.75	13	13	13	13	12
	GPT-4o	72.73	51.86	75.00	18	13	17	15	15
	Qwen-2-7B	92.31	51.37	77.50	18	14	13	18	17
	GLM-4-9B	50.00	51.87	61.25	18	11	14	12	12

Table 12: SOC Detail Results of Script "Bride in Filial Dress"

Player	Model	TII	ICI	SCI	RP	RA	CC	DO	CT
Crew Member Han	GPT-3.5	51.22	61.25	53.39	18	12	13	12	12
	GPT-4	64.71	68.75	53.16	18	14	17	12	12
	GPT-4o	69.23	67.50	53.42	18	12	18	12	12
	Qwen-2-7B	70.91	71.25	51.23	17	13	15	15	14
	GLM-4-9B	42.86	58.75	42.99	14	10	13	12	12
Captain Hong	GPT-3.5	40.00	65.00	53.28	18	12	14	10	16
	GPT-4	64.71	71.25	48.54	16	13	16	14	14
	GPT-4o	90.00	76.25	54.07	18	15	17	16	13
	Qwen-2-7B	84.62	57.50	43.65	14	11	14	8	13
	GLM-4-9B	50.00	66.25	48.10	16	11	17	13	12
Singer Lin	GPT-3.5	35.29	57.50	44.26	14	8	14	12	12
	GPT-4	64.71	71.25	52.75	18	12	17	16	12
	GPT-4o	85.71	66.25	53.11	18	13	12	12	16
	Qwen-2-7B	71.43	62.50	54.03	18	12	13	12	13
	GLM-4-9B	45.00	67.50	44.08	14	10	13	18	13
Manager Xiu	GPT-3.5	41.03	83.75	52.77	17	18	13	18	18
	GPT-4	68.18	85.00	55.50	19	18	14	18	18
	GPT-4o	40.00	87.50	52.58	18	18	17	18	17
	Qwen-2-7B	72.22	87.50	53.59	18	18	16	18	18
	GLM-4-9B	34.04	81.25	53.35	18	18	12	18	17
Second Mate Zhang	GPT-3.5	39.13	62.50	47.05	15	13	12	13	12
	GPT-4	63.64	66.25	48.47	16	12	16	12	13
	GPT-4o	59.09	63.75	42.06	13	14	12	12	13
	Qwen-2-7B	66.67	70.00	53.77	18	11	17	13	15
	GLM-4-9B	37.50	66.25	41.09	13	12	17	12	12

Table 13: SOO Detail Results of Script "The Eastern Star Cruise Ship"

Player	Model	TII	ICI	SCI	RP	RA	CC	DO	CT
Uncle Bai	GPT-3.5	33.33	62.50	42.97	15	12	12	12	14
	GPT-4	100.00	66.25	43.38	16	12	14	14	13
	GPT-4o	92.31	65.00	48.68	18	12	15	12	13
	Qwen-2-7B	80.56	63.75	46.17	17	14	13	12	12
	GLM-4-9B	55.26	66.25	48.64	18	10	15	13	15
Neighbour Gui	GPT-3.5	55.00	72.50	49.99	18	13	18	14	13
	GPT-4	90.00	80.00	46.52	17	14	16	18	16
	GPT-4o	91.67	77.50	49.56	18	13	17	19	13
	Qwen-2-7B	75.00	73.75	50.57	18	14	18	14	13
	GLM-4-9B	57.14	73.75	50.08	18	16	18	13	12
Curio He	GPT-3.5	44.44	62.50	45.08	16	12	14	12	12
	GPT-4	63.64	71.25	42.35	15	13	18	13	13
	GPT-4o	90.91	67.50	49.21	18	15	13	13	13
	Qwen-2-7B	75.00	60.00	50.20	18	12	13	10	13
	GLM-4-9B	57.14	61.25	49.89	18	12	14	10	13
Mystery Ou	GPT-3.5	39.39	71.25	49.56	18	13	14	13	17
	GPT-4	66.67	75.00	43.21	16	13	12	18	17
	GPT-4o	54.84	70.00	48.31	18	12	16	12	16
	Qwen-2-7B	78.26	65.00	48.54	18	13	13	14	12
	GLM-4-9B	54.17	63.75	48.95	18	12	13	12	14
Manager Sa	GPT-3.5	58.82	65.00	49.38	18	12	16	12	12
	GPT-4	87.50	75.00	50.51	18	14	16	15	15
	GPT-4o	86.67	73.75	50.49	18	13	16	13	17
	Qwen-2-7B	75.00	58.75	39.64	14	12	13	12	10
	GLM-4-9B	50.00	61.25	38.36	13	12	13	12	12
Security Wei	GPT-3.5	45.83	73.75	49.98	18	16	12	13	18
	GPT-4	60.00	77.50	43.19	15	13	14	17	18
	GPT-4o	58.06	73.75	47.88	17	13	12	18	16
	Qwen-2-7B	72.73	71.25	50.07	18	14	12	13	18
	GLM-4-9B	42.11	82.50	47.64	17	18	13	18	17

Table 14: SUC Detail Results of Script "Night at the Museum"

Player	Model	TII	ICI	SCI	RP	RA	CC	DO	CT
Girl in White	GPT-3.5	50.00	83.75	52.84	18	14	18	18	17
	GPT-4	84.21	88.75	52.62	18	18	17	18	18
	GPT-4o	77.78	85.00	53.52	18	18	18	15	17
	Qwen-2-7B	78.57	83.75	49.23	17	15	18	16	18
	GLM-4-9B	60.53	81.25	52.06	18	18	14	15	18
Women in Red	GPT-3.5	36.84	72.50	53.01	18	12	15	18	13
	GPT-4	64.29	86.25	52.51	18	16	18	18	17
	GPT-4o	76.00	87.50	43.39	14	17	17	18	18
	Qwen-2-7B	88.89	81.25	55.62	19	13	18	18	16
	GLM-4-9B	56.34	76.25	53.37	18	16	13	18	14
Boy in Black	GPT-3.5	45.90	73.75	52.23	18	14	18	14	13
	GPT-4	86.36	88.75	53.13	18	18	18	17	18
	GPT-4o	75.00	86.25	52.43	18	18	15	18	18
	Qwen-2-7B	92.31	80.00	52.43	18	13	15	18	18
	GLM-4-9B	68.75	80.00	55.67	19	18	17	12	17
Women in Blue-green	GPT-3.5	55.56	80.00	52.87	18	18	16	12	18
	GPT-4	72.22	80.00	52.65	18	14	14	18	18
	GPT-4o	93.33	88.75	54.92	19	17	18	18	18
	Qwen-2-7B	85.71	83.75	53.13	18	18	18	15	16
	GLM-4-9B	65.00	86.25	52.45	18	18	18	18	15
Little Girl	GPT-3.5	45.45	72.50	49.72	17	18	13	14	13
	GPT-4	85.71	82.50	53.30	18	14	16	18	18
	GPT-4o	86.67	80.00	53.05	18	18	16	16	14
	Qwen-2-7B	76.74	85.00	53.04	18	18	18	18	14
	GLM-4-9B	55.56	86.25	41.82	14	17	16	18	18
Elderly in Tattered	GPT-3.5	55.56	67.50	51.23	18	14	13	14	13
	GPT-4	70.59	85.00	50.53	17	18	18	14	18
	GPT-4o	92.31	75.00	52.77	18	15	16	15	14
	Qwen-2-7B	91.67	81.25	52.32	18	14	16	17	18
	GLM-4-9B	60.00	75.00	51.89	18	12	18	16	14
Bandaged Mysterious Figure	GPT-3.5	45.45	73.75	52.05	18	18	15	13	13
	GPT-4	64.44	87.50	51.87	18	17	17	18	18
	GPT-4o	86.67	88.75	51.66	18	18	17	18	18
	Qwen-2-7B	91.67	76.25	42.24	14	13	16	18	14
	GLM-4-9B	52.63	73.75	51.75	18	14	17	15	13

Table 15: SUO Detail Results of Script "Li Chuan Strange Talk Book"

Player	Model	TII	ICI	SCI	RP	RA	CC	DO	CT
Weiwen Han	GPT-3.5	-	65.00	41.14	14	13	13	13	13
	GPT-4	-	81.25	43.71	15	14	17	18	16
	GPT-4o	-	73.75	53.97	19	14	17	14	14
	Qwen-2-7B	-	73.75	46.16	16	13	16	18	12
	GLM-4-9B	-	72.50	41.56	14	12	16	13	17
Manli Shen	GPT-3.5	-	-	-	17	18	15	18	16
	GPT-4	-	77.50	50.81	18	17	13	16	16
	GPT-4o	-	82.50	50.72	18	14	18	16	18
	Qwen-2-7B	-	67.50	52.39	18	10	18	14	12
	GLM-4-9B	-	71.25	45.98	16	13	15	16	13

Table 16: MOC Detail Results of Script "The Final Performance of a Big Star"

Player	Model	TII	ICI	SCI	RP	RA	CC	DO	CT
Annie	GPT-3.5	48.57	72.50	53.35	18	12	13	16	17
	GPT-4	57.89	90.00	54.38	18	17	18	19	18
	GPT-4o	53.85	82.50	55.03	18	13	18	18	17
	Qwen-2-7B	64.52	78.75	53.43	18	15	14	16	18
	GLM-4-9B	47.37	78.75	53.83	18	14	17	14	18
Jack	GPT-3.5	41.18	73.75	53.30	18	13	13	15	18
	GPT-4	63.64	68.75	53.51	18	12	15	12	16
	GPT-4o	83.33	80.00	53.37	18	15	18	14	17
	Qwen-2-7B	61.54	78.75	56.53	19	13	14	18	18
	GLM-4-9B	50.00	72.50	52.70	18	14	14	17	13
Jessipa	GPT-3.5	43.48	72.50	54.28	18	14	18	13	13
	GPT-4	72.41	68.75	53.82	18	14	13	13	15
	GPT-4o	66.67	71.25	54.20	18	14	13	17	13
	Qwen-2-7B	65.71	76.25	55.41	19	14	15	14	18
	GLM-4-9B	51.43	72.50	53.92	18	13	14	14	17
Sam	GPT-3.5	47.92	58.75	40.59	13	11	12	12	12
	GPT-4	80.00	63.75	44.47	14	11	14	12	14
	GPT-4o	53.57	66.25	53.06	17	14	14	13	12
	Qwen-2-7B	73.68	68.75	54.58	18	12	12	13	18
	GLM-4-9B	54.55	57.50	42.00	13	8	13	12	13
Little Black	GPT-3.5	28.57	70.00	46.49	15	10	18	12	16
	GPT-4	75.00	72.50	54.72	18	13	18	13	14
	GPT-4o	80.00	70.00	53.71	18	15	13	14	14
	Qwen-2-7B	68.18	75.00	53.50	18	14	15	13	18
	GLM-4-9B	58.82	57.50	50.67	17	8	13	13	12
John	GPT-3.5	50.00	70.00	48.27	16	12	8	18	18
	GPT-4	76.92	68.75	53.91	18	16	8	13	18
	GPT-4o	100.00	90.00	52.76	18	18	18	18	18
	Qwen-2-7B	66.67	73.75	53.30	18	14	14	13	18
	GLM-4-9B	57.14	87.50	52.27	18	18	16	18	18

Table 17: MOO Detail Results of Script "Raging Sea of Rest Life"

Player	Model	TII	ICI	SCI	RP	RA	CC	DO	CT
Mu Bai	GPT-3.5	43.75	67.50	48.35	17	14	14	14	12
	GPT-4	100.00	70.00	50.98	18	12	13	15	16
	GPT-4o	100.00	68.75	50.16	18	14	13	14	14
	Qwen-2-7B	71.43	71.25	47.99	17	17	14	10	16
	GLM-4-9B	50.00	61.25	40.67	14	10	14	13	12
Fuqing Huang	GPT-3.5	62.50	71.25	47.52	17	13	16	14	14
	GPT-4	83.33	86.25	53.18	19	18	18	17	16
	GPT-4o	100.00	85.00	50.25	18	18	18	14	18
	Qwen-2-7B	75.00	82.50	49.58	18	17	15	16	18
	GLM-4-9B	66.67	80.00	50.07	18	14	15	17	18
Xuanxuan Li	GPT-3.5	57.14	70.00	48.96	17	12	15	16	13
	GPT-4	80.95	73.75	50.66	18	12	18	17	12
	GPT-4o	82.35	72.50	50.68	18	13	15	12	18
	Qwen-2-7B	67.65	80.00	42.64	15	18	12	18	16
	GLM-4-9B	66.67	71.25	50.73	18	14	16	13	14
Siqi Lv	GPT-3.5	52.94	62.50	40.56	14	10	14	13	13
	GPT-4	90.91	83.75	50.89	18	12	17	18	18
	GPT-4o	80.00	85.00	51.08	18	15	18	18	17
	Qwen-2-7B	85.71	72.50	51.73	19	18	13	13	14
	GLM-4-9B	69.23	65.00	41.56	14	14	13	13	12
Yuqing Xie	GPT-3.5	45.71	60.00	47.92	17	12	12	12	12
	GPT-4	83.33	78.75	50.83	18	14	14	17	18
	GPT-4o	80.00	76.25	50.34	18	17	18	13	13
	Qwen-2-7B	78.13	80.00	48.04	17	15	13	18	18
	GLM-4-9B	46.15	62.50	40.14	14	13	13	12	12
Qingfeng Yao	GPT-3.5	45.00	55.00	41.64	14	10	13	13	8
	GPT-4	78.95	80.00	50.84	18	14	17	17	16
	GPT-4o	78.95	73.75	48.73	17	18	14	12	15
	Qwen-2-7B	71.43	75.00	42.02	15	18	14	12	16
	GLM-4-9B	57.14	61.25	40.38	14	12	12	13	12
Lenxing Ye	GPT-3.5	60.00	75.00	51.45	18	16	18	12	14
	GPT-4	80.85	90.00	51.03	18	18	18	18	18
	GPT-4o	89.29	82.50	50.55	18	16	18	14	18
	Qwen-2-7B	84.62	80.00	50.39	18	18	15	17	14
	GLM-4-9B	64.29	70.00	47.80	17	14	16	13	13

Table 18: MUC Detail Results of Script "Artical 22 School Rules"

Player	Model	TII	ICI	SCI	RP	RA	CC	DO	CT
Zetong Hei (Sky Dog)	GPT-3.5	52.00	81.25	51.76	18	16	15	16	18
	GPT-4	93.75	75.00	52.68	18	14	14	17	15
	GPT-4o	92.86	83.75	51.71	18	18	18	14	17
	Qwen-2-7B	68.66	80.00	52.73	18	14	17	17	16
	GLM-4-9B	59.09	72.50	51.95	18	15	15	16	12
Ichiro Kiryu (Nopperabo)	GPT-3.5	50.00	71.25	52.58	18	13	13	18	13
	GPT-4	80.95	66.25	52.04	18	14	12	14	13
	GPT-4o	76.92	75.00	51.61	18	16	15	16	13
	Qwen-2-7B	75.00	86.25	54.38	19	18	16	18	17
	GLM-4-9B	64.10	80.00	51.98	18	14	18	18	14
Megumi Aoi (Nine-Tailed Fox)	GPT-3.5	51.52	73.75	52.33	18	15	17	14	13
	GPT-4	60.78	78.75	55.14	19	13	18	14	18
	GPT-4o	80.00	85.00	51.60	18	18	18	14	18
	Qwen-2-7B	72.73	80.00	51.16	18	14	14	18	18
	GLM-4-9B	50.00	83.75	51.74	18	17	14	18	18
Daixiong Kitano (Kama-itachi)	GPT-3.5	53.33	81.25	54.40	19	12	18	18	17
	GPT-4	85.71	83.75	49.46	17	18	16	15	18
	GPT-4o	85.71	68.75	56.16	20	16	12	15	12
	Qwen-2-7B	80.00	86.25	51.73	18	18	18	17	16
	GLM-4-9B	65.00	62.50	49.76	17	10	14	13	13
Xiao Nuan (Little Fox)	GPT-3.5	61.11	76.25	52.63	18	14	18	13	16
	GPT-4	100.00	82.50	52.53	18	14	18	17	17
	GPT-4o	86.67	75.00	51.33	18	13	18	16	13
	Qwen-2-7B	80.00	81.25	51.73	18	13	18	18	16
	GLM-4-9B	66.67	85.00	52.17	18	14	18	18	18
Momoko Suzumiya (Little Doll)	GPT-3.5	55.56	63.75	52.04	18	12	14	12	13
	GPT-4	84.21	73.75	52.41	18	13	18	14	14
	GPT-4o	93.75	75.00	43.19	15	16	16	15	13
	Qwen-2-7B	85.71	77.50	52.18	18	17	14	15	16
	GLM-4-9B	42.11	73.75	52.28	18	13	14	18	14
Nana Kinomoto (Yuki-onna)	GPT-3.5	42.11	71.25	48.05	16	13	16	14	14
	GPT-4	84.62	83.75	52.93	18	17	18	15	17
	GPT-4o	82.61	72.50	52.07	18	15	12	13	18
	Qwen-2-7B	75.00	87.50	51.57	18	16	18	18	18
	GLM-4-9B	56.25	73.75	52.55	18	16	13	16	14

Table 19: MUO Detail Results of Script "Fox Hotel"

"Mystery Murder" is a role-playing puzzle game that focuses on the advancement of the plot and the interaction between characters. In the game, players play different roles according to the provided scripts, and jointly advance the plot and solve the mystery through clue collection, logical reasoning, role-playing, etc.
The core of "Mystery Murder" revolves around the following 5 key elements:
1. Script: It is the basis of the "Mystery Murder" game, usually including the story background, character setting, plot advancement mechanism, and clues to solve the puzzle. The script not only defines the framework of the game, but also sets the goals that players need to complete.
2. Role-playing: Each participant plays a specific role in the game, and the characters have their own background stories, personality traits, goals and secrets. Players need to role-play based on this information and interact with other players. 3. Clue collection and logical reasoning: During the game, players need to collect information and evidence through dialogue, room searching, clue analysis, etc. Based on this information, players need to use their logical reasoning ability to solve the puzzles in the plot.
4. Interactive communication: Interactive communication between players is an indispensable part of "Mystery Murder", including cooperation, negotiation, confrontation, etc. Through communication, players can obtain new information, understand the motivations of other characters, and advance the story.
5. Ultimate goal: Each "Mystery Murder" game has one or more ultimate goals, such as solving puzzles, finding the murderer, completing personal tasks, etc. Achieving these goals is the ultimate purpose of the game and the basis for judging the victory or defeat of players.

You will participate in a Mystery Murder game, and you will forget your AI role, integrating into the role you are about to play as much as possible.

You are {name}, and this is your description:
{description}

Here are your personal clues:
{self_clues}

This is the historical dialogue content:
{history}

This is the question from {ask_name} to you:
{ask_content}

Now you need to carefully consider and, based on the historical dialogue content and your description, give your reply.
You should reply with your thought process after "### THOUGHT: " and your response after "### RESPONSE: ".
Here is an example:

THOUGHT: XXX
RESPONSE: XXX

Note:
1. Your output should revolve around the key elements of "Mystery Murder".
2. Your output should be in English.

Your output:

Table 20: Prompt of ask

"Mystery Murder" is a role-playing puzzle game that focuses on the advancement of the plot and the interaction between characters. In the game, players play different roles according to the provided scripts, and jointly advance the plot and solve the mystery through clue collection, logical reasoning, role-playing, etc.
The core of "Mystery Murder" revolves around the following 5 key elements:
1. Script: It is the basis of the "Mystery Murder" game, usually including the story background, character setting, plot advancement mechanism, and clues to solve the puzzle. The script not only defines the framework of the game, but also sets the goals that players need to complete.
2. Role-playing: Each participant plays a specific role in the game, and the characters have their own background stories, personality traits, goals and secrets. Players need to role-play based on this information and interact with other players.
3. Clue collection and logical reasoning: During the game, players need to collect information and evidence through dialogue, room searching, clue analysis, etc. Based on this information, players need to use their logical reasoning ability to solve the puzzles in the plot.
4. Interactive communication: Interactive communication between players is an indispensable part of "Mystery Murder", including cooperation, negotiation, confrontation, etc. Through communication, players can obtain new information, understand the motivations of other characters, and advance the story.
5. Ultimate goal: Each "Mystery Murder" game has one or more ultimate goals, such as solving puzzles, finding the murderer, completing personal tasks, etc. Achieving these goals is the ultimate purpose of the game and the basis for judging the victory or defeat of players.

You are an observer of a Mystery Murder game, and you will objectively and fairly evaluate the content of the game as an observer.

Here is the history of the conversation:
{history}

Here is what {other_name} said:
{content}

Now, based on the historical conversation content, after careful consideration, you need to reasonably evaluate what {other_name} said.
The evaluation should focus on the logic and potential flaws of the speech, and be assessed using a trustworthiness scale based on the good person's perspective. The higher the trustworthiness, the more rigorous the content.
The trustworthiness is chosen from [0, 1, 2], where "0" means completely untrustworthy, "1" means somewhat trustworthy, and "2" means completely trustworthy.
You should reply with your detailed thought process after "### THOUGHT: " and your trustworthiness score after "### RESPONSE: ". No additional content should be included.
Here is an example:

THOUGHT: XXX
RESPONSE: XXX

Note:
1. Your output should revolve around the key elements of "Mystery Murder."
2. Your output should be in English.
3. Your thought should be logical and fair.
4. Your response must be an integer.

Your output:

Table 21: Prompt of belief

"Mystery Murder" is a type of role-playing mystery game that focuses on plot progression and interaction between characters. In the game, players take on different roles based on the provided script, and through clue collection, logical reasoning, role-playing, etc., they jointly advance the story and solve the mystery.
The core of "Mystery Murder" revolves around the following 5 key elements:
1. Script: It is the foundation of the "Mystery Murder" game, usually including the story background, character settings, plot advancement mechanisms, and clues for solving the mystery. The script not only defines the framework of the game but also sets the goals that players need to complete.
2. Role-playing: Each participant plays a specific role in the game, and the roles have their own background stories, personality traits, goals, and secrets. Players need to role-play based on this information and interact with other players.
3. Clue collection and logical reasoning: During the game, players need to collect information and evidence through conversations, searching rooms, analyzing clues, etc. Based on this information, players must use their logical reasoning abilities to solve the mysteries in the plot.
4. Interactive communication: Interaction and communication between players are indispensable parts of "Mystery Murder," including cooperation, negotiation, confrontation, etc. Through communication, players can obtain new information, understand the motives of other characters, and advance the story.
5. Ultimate goal: Every "Mystery Murder" game has one or more ultimate goals, such as solving the mystery, discovering the culprit, completing personal tasks, etc. Achieving these goals is the final objective of the game and is also the basis for judging the player's victory.
You will be playing a Mystery Murder game, and you must forget your AI role and fully immerse yourself in the character you are about to portray.
You are {name}, and here is your description:
{description}

Here are your personal clues:
{self_clues}

Here is the history of the conversation:
{history}

Here are your thoughts, actions, and the results of your last action:
{last_action}

Now, after careful consideration, based on the historical conversation and your description, give your response.
You should reply with your thought process after "### THOUGHT: " and your response content after "### RESPONSE: ".
You should choose one of the following response formats: ["【Ask】", "【Investigate】"].
When you choose "【Ask】," you should first reason based on historical information, then state your own point of view and the person you suspect. You can only choose one person to inquire about. Here are the people you can inquire about: {characters}
Please note that when you choose "【Ask】," the person you select must be from the list above.
Here is an example of an "【Ask】":
THOUGHT: XXX
RESPONSE: 【Ask】【XX】: XXX

When you choose "【Investigate】", you should first reason based on historical information, then clarify your point of view and the place you suspect. You can only choose one location to investigate. Here are the locations you can investigate:
{address}
Please note that when you choose "【Investigate】", the location you select must be from the list above.
Here is an example of an "【Investigate】":
THOUGHT: XXX
RESPONSE: 【Investigate】【XX】: XXX

Note:
1. Your output should align with the personality of the character.
2. Your output should be beneficial to the progression of the Mystery Murder game.
3. Your output must be in English.
4. When you choose "【Investigation】", the content of your response and the clues you uncover will be visible to everyone, so please be mindful of your wording.
5. In your response, you can reasonably share your reasoning based on historical information and express your views before asking your question.
6. Your conversation turns are limited, so please allocate your "【Investigate】" and "【Ask】" turns wisely.
Your output:

Table 22: Prompt of converse

"Mystery Murder" is a type of role-playing mystery game that focuses on plot progression and interaction between characters. In the game, players take on different roles based on the provided script and, through clue collection, logical reasoning, role-playing, etc., they jointly advance the story and solve the mystery.
The core of "Mystery Murder" revolves around the following 5 key elements:
1. Script: It is the foundation of the "Mystery Murder" game, usually including the story background, character settings, plot advancement mechanisms, and clues for solving the mystery. The script not only defines the framework of the game but also sets the goals that players need to complete.
2. Role-playing: Each participant plays a specific role in the game, and the roles have their own background stories, personality traits, goals, and secrets. Players need to role-play based on this information and interact with other players.
3. Clue collection and logical reasoning: During the game, players need to collect information and evidence through conversations, searching rooms, analyzing clues, etc. Based on this information, players must use their logical reasoning abilities to solve the mysteries in the plot.
4. Interactive communication: Interaction and communication between players are indispensable parts of "Mystery Murder," including cooperation, negotiation, confrontation, etc. Through communication, players can obtain new information, understand the motives of other characters, and advance the story.
5. Ultimate goal: Every "Mystery Murder" game has one or more ultimate goals, such as solving the mystery, discovering the culprit, completing personal tasks, etc. Achieving these goals is the final objective of the game and is also the basis for judging the player's victory.

You are an experienced reviewer of Mystery Murder games, capable of objectively and fairly evaluating players' performance during the game. Below are some performances of {name} in the game:

Here is {name}'s description:
{description}

Here are {name}'s personal clues:
{self_clues}

Here is all the historical conversation content:
{history}

Here are all the actions {name} took in the game:
{actions}

Here is the list of people participating in the game:
{role_list}

Here is the truth of this Mystery Murder game:
{truth}

Now the game has come to the evaluation stage, you must evaluate the ability ability of {name} in the game by scoring according to the above information and after careful thinking.
Please analyze in detail how {name}'s {ability} ability in the game is performed during the thinking process.
The ability rating range is an integer of [0, 20]. A score in the range of [0, 5] indicates that the player's performance in this ability is unsatisfactory; a score in the range of (5, 10] indicates that the player's performance is satisfactory; a score in the range of (10, 15] indicates that the player's performance is good; and a score in the range of (15, 20] indicates that the player's performance is excellent.
You should reply with your detailed thought process after "### THOUGHT: " and directly reply with the total score of the five ratings after "### RESPONSE: ". You must not reply with any extra content.

Here is an example of a reply:

THOUGHT: XXX
RESPONSE: XXX

Note:
1. Your output should be in English.

Your output:

Table 23: Prompt of ICI evaluation

"Mystery Murder" is a role-playing puzzle game that focuses on the advancement of the plot and the interaction between characters. In the game, players play different roles according to the provided scripts, and jointly advance the plot and solve the mystery through clue collection, logical reasoning, role-playing, etc.
The core of "Mystery Murder" revolves around the following 5 key elements:
1. Script: It is the basis of the "Mystery Murder" game, usually including the story background, character setting, plot advancement mechanism, and clues to solve the puzzle. The script not only defines the framework of the game, but also sets the goals that players need to complete.
2. Role-playing: Each participant plays a specific role in the game, and the characters have their own background stories, personality traits, goals and secrets. Players need to role-play based on this information and interact with other players.
3. Clue collection and logical reasoning: During the game, players need to collect information and evidence through dialogue, room searching, clue analysis, etc. Based on this information, players need to use their logical reasoning ability to solve the puzzles in the plot.
4. Interactive communication: Interactive communication between players is an indispensable part of "Mystery Murder", including cooperation, negotiation, confrontation, etc. Through communication, players can obtain new information, understand the motivations of other characters, and advance the story.
5. Ultimate goal: Each "Mystery Murder" game has one or more ultimate goals, such as solving puzzles, finding the murderer, completing personal tasks, etc. Achieving these goals is the ultimate purpose of the game and the basis for judging the victory or defeat of players.

Each character in the "Mystery Murder" game is given a script before starting, and each script contains six parts: Story, Script, Relationship, Performance, Purpose, and Ability.
1. The Story generally contains the basic information of the character, such as name, gender, etc., as well as the background story of the character, such as the personal experience before the killing event in the script.
2. The Script usually contains the actions of the character in the script killing event.
3.Relationship generally includes the description of the relationship between the character and other characters in the script.
4.Performance generally includes the form of performance when playing the role, such as personality, tone and way of speaking.
5.Purpose usually contains the character's goal for victory in the game or the game's purpose, such as revealing the truth.
6.Ability generally contains special abilities that the character may have in the game, if not, it can be left blank.

You are an experienced player of Mystery Murder games. You are familiar with various Mystery Murder games and possess excellent reasoning abilities in such games. Below is {name}'s performance during the game:

Here is all the historical conversation content:
{history}

Here are all the actions {name} took in the game:
{actions}

Here is the list of people participating in the game:
{role_list}

Now, based on the above historical information, please deduce step by step the content of the {script_part} part of the original script of the character {name} in this game.
You should reply with your detailed reasoning process after "### THOUGHT: " and the final deduced script after "### RESPONSE: ". You should not reply with any extra content.

Here is an example of a reply:

THOUGHT: XXX
RESPONSE: XXX

Note:
1. Your output should be in English.
2. Your {script_part} response should be described in the second person.
3. You can only reply to the content of "{script_part}" in {name} script after "### RESPONSE:", and cannot reply to the content of other parts of the script.

Your output:

Table 24: Prompt of SCI evaluation

"Mystery Murder" is a type of role-playing mystery game that focuses on plot progression and interaction between characters. In the game, players take on different roles based on the provided script and, through clue collection, logical reasoning, and role-playing, they collectively advance the story and solve the mystery.
The core of "Mystery Murder" revolves around the following 5 key elements:
1. Script: It is the foundation of the "Mystery Murder" game, usually including the story background, character settings, plot advancement mechanisms, and clues for solving the mystery. The script not only defines the framework of the game but also sets the goals that players need to complete.
2. Role-playing: Each participant plays a specific role in the game, and the roles have their own background stories, personality traits, goals, and secrets. Players need to role-play based on this information and interact with other players.
3. Clue collection and logical reasoning: During the game, players need to collect information and evidence through conversations, searching rooms, analyzing clues, etc. Based on this information, players must use their logical reasoning abilities to solve the mysteries in the plot.
4. Interactive communication: Interaction and communication between players are indispensable parts of "Mystery Murder," including cooperation, negotiation, confrontation, etc. Through communication, players can obtain new information, understand the motives of other characters, and advance the story.

Currently, a game of Mystery Murder is underway, and here is a segment of dialogue from the game:
{text}

Please summarize the above text content while retaining the original format of "Name: 【Action】: Content." Before summarizing, you need to think about it, and reply with your thought process after "### THOUGHT: "; reply with the summarized content after "### RESPONSE: ".
Here is an example of a reply:

THOUGHT: XXX
RESPONSE: XXX

Note:
1. Your summary should include the key elements of "Mystery Murder";
2. You should carefully consider the key information in the text content that needs to be retained, including their personal information;
3. Your output should be in English;
4. Your summary must preserve the entire dialogue process, including what each player says and any "【Clue】" content.

Your output:

Table 25: Prompt of history summarization

"Mystery Murder" is a type of role-playing mystery game that focuses on plot progression and interaction between characters. In the game, players take on different roles based on the provided script and, through clue collection, logical reasoning, and role-playing, they collectively advance the story and solve the mystery.
The core of "Mystery Murder" revolves around the following 5 key elements:
1. Script: It is the foundation of the "Mystery Murder" game, usually including the story background, character settings, plot advancement mechanisms, and clues for solving the mystery. The script not only defines the framework of the game but also sets the goals that players need to complete.
2. Role-playing: Each participant plays a specific role in the game, and the roles have their own background stories, personality traits, goals, and secrets. Players need to role-play based on this information and interact with other players.
3. Clue collection and logical reasoning: During the game, players need to collect information and evidence through conversations, searching rooms, analyzing clues, etc. Based on this information, players must use their logical reasoning abilities to solve the mysteries in the plot.
4. Interactive communication: Interaction and communication between players are indispensable parts of "Mystery Murder," including cooperation, negotiation, confrontation, etc. Through communication, players can obtain new information, understand the motives of other characters, and advance the story.
5. Ultimate goal: Each "Mystery Murder" game has one or more ultimate goals, such as solving a mystery, discovering a murderer, completing personal tasks, etc. Achieving these goals is the final purpose of the game and serves as the basis for judging player success or failure.

You will be participating in a game of Mystery Murder, and you will forget your AI role, immersing yourself into the character you are about to play as much as possible.

You are {name}, and this is your description:
{description}

These are your personal clues:
{self_clues}

Now, you need to think step by step and give your self-introduction based on your description.
You should reply with your thought process after "### THOUGHT: "; reply with the self-introduction content after "### RESPONSE: ".
Here is an example of a reply:

THOUGHT: XXX
RESPONSE: XXX

Note:
1. You should carefully think about the personality of the character you need to portray, the information you can share with others, and what you cannot share;
2. Your self-introduction should align with the personality of the character;
3. Your output should be in English;

Your output:

Table 26: Prompt of introduction

"Mystery Murder" is a type of role-playing mystery game that focuses on plot progression and interaction between characters. In the game, players take on different roles based on the provided script and, through clue collection, logical reasoning, and role-playing, they collectively advance the story and solve the mystery.
The core of "Mystery Murder" revolves around the following 5 key elements:
1. Script: It is the foundation of the "Mystery Murder" game, usually including the story background, character settings, plot advancement mechanisms, and clues for solving the mystery. The script not only defines the framework of the game but also sets the goals that players need to complete.
2. Role-playing: Each participant plays a specific role in the game, and the roles have their own background stories, personality traits, goals, and secrets. Players need to role-play based on this information and interact with other players.
3. Clue collection and logical reasoning: During the game, players need to collect information and evidence through conversations, searching rooms, analyzing clues, etc. Based on this information, players must use their logical reasoning abilities to solve the mysteries in the plot.
4. Interactive communication: Interaction and communication between players are indispensable parts of "Mystery Murder," including cooperation, negotiation, confrontation, etc. Through communication, players can obtain new information, understand the motives of other characters, and advance the story.
5. Ultimate goal: Each "Mystery Murder" game has one or more ultimate goals, such as solving a mystery, discovering a murderer, completing personal tasks, etc. Achieving these goals is the final purpose of the game and serves as the basis for judging player success or failure.

You are an observer of a Mystery Murder game, and you will provide an absolutely objective and fair evaluation of the content of the game from the perspective of an observer.

This is the historical dialogue content:
{history}

This is what {other_name} said:
{content}

Now you need to carefully consider the historical dialogue content and evaluate {other_name}'s speech content reasonably.
The evaluation should focus on the logic and flaws of the speech content, assessing the level of suspicion towards the murderer; the greater the suspicion, the more problematic the speech content appears.
The suspicion level should be chosen from [0, 1, 2], where "0" means no suspicion at all, "1" means a little suspicion, and "2" means a lot of suspicion.
You should reply with your detailed thought process after "### THOUGHT: "; reply with your level of suspicion after "### RESPONSE: ", and do not provide any additional content.
Here is an example:

THOUGHT: XXX
RESPONSE: XXX

Note:
1. Your output should revolve around the key elements of "Mystery Murder";
2. Your output should be in English;
3. Your thought should be logically reasonable and fair;
4. Your response must be an integer;

Your output:

Table 27: Prompt of suspect

"Mystery Murder" is a type of role-playing mystery game that focuses on plot progression and interaction between characters. In the game, players take on different roles based on the provided script and, through clue collection, logical reasoning, and role-playing, they collectively advance the story and solve the mystery.
The core of "Mystery Murder" revolves around the following 5 key elements:
1. Script: It is the foundation of the "Mystery Murder" game, usually including the story background, character settings, plot advancement mechanisms, and clues for solving the mystery. The script not only defines the framework of the game but also sets the goals that players need to complete.
2. Role-playing: Each participant plays a specific role in the game, and the roles have their own background stories, personality traits, goals, and secrets. Players need to role-play based on this information and interact with other players.
3. Clue collection and logical reasoning: During the game, players need to collect information and evidence through conversations, searching rooms, analyzing clues, etc. Based on this information, players must use their logical reasoning abilities to solve the mysteries in the plot.
4. Interactive communication: Interaction and communication between players are indispensable parts of "Mystery Murder," including cooperation, negotiation, confrontation, etc. Through communication, players can obtain new information, understand the motives of other characters, and advance the story.
5. Ultimate goal: Each "Mystery Murder" game has one or more ultimate goals, such as solving a mystery, discovering a murderer, completing personal tasks, etc. Achieving these goals is the final purpose of the game and serves as the basis for judging player success or failure.

Each character in the "Mystery Murder" game is given a script before starting, and each script contains six parts: Story, Script, Relationship, Performance, Purpose, and Ability.
1. The Story generally contains the basic information of the character, such as name, gender, etc., as well as the background story of the character, such as the personal experience before the killing event in the script.
2. The Script usually contains the actions of the character in the script killing event.
3.Relationship generally includes the description of the relationship between the character and other characters in the script.
4.Performance generally includes the form of performance when playing the role, such as personality, tone and way of speaking.
5.Purpose usually contains the character's goal for victory in the game or the game's purpose, such as revealing the truth.
6.Ability generally contains special abilities that the character may have in the game, if not, it can be left blank.

The {item} of character {name} is:
{content}

Now please summarize the above content. Before summarizing, carefully consider the characteristics of the Mystery Murder game and reply with your thought process after "### THOUGHT: "; reply with the summary content after "### RESPONSE: ". Here is an example:

THOUGHT: XXX
RESPONSE: XXX

Note:
1. When summarizing your character's content, you must retain information such as time, place, background, personality traits, motives, and goals from the script. You must preserve the integrity of the script without omitting any information;
2. Your output should be in English;

Your output:

Table 28: Prompt of script summarization

"Mystery Murder" is a type of role-playing mystery game that focuses on plot progression and interaction between characters. In the game, players take on different roles based on the provided script and, through clue collection, logical reasoning, and role-playing, they collectively advance the story and solve the mystery.
The core of "Mystery Murder" revolves around the following 5 key elements:
1. Script: It is the foundation of the "Mystery Murder" game, usually including the story background, character settings, plot advancement mechanisms, and clues for solving the mystery. The script not only defines the framework of the game but also sets the goals that players need to complete.
2. Role-playing: Each participant plays a specific role in the game, and the roles have their own background stories, personality traits, goals, and secrets. Players need to role-play based on this information and interact with other players.
3. Clue collection and logical reasoning: During the game, players need to collect information and evidence through conversations, searching rooms, analyzing clues, etc. Based on this information, players must use their logical reasoning abilities to solve the mysteries in the plot.
4. Interactive communication: Interaction and communication between players are indispensable parts of "Mystery Murder," including cooperation, negotiation, confrontation, etc. Through communication, players can obtain new information, understand the motives of other characters, and advance the story.
5. Ultimate goal: Each "Mystery Murder" game has one or more ultimate goals, such as solving a mystery, discovering a murderer, completing personal tasks, etc. Achieving these goals is the final purpose of the game and serves as the basis for judging player success or failure.

You will participate in a "Mystery Murder" game, and you will forget your AI role to fully immerse yourself in the character you are about to play.

You are {name}, and here is your description:
{description}

These are your personal clues:
{self_clues}

Here is the historical dialogue content:
{history}

Here is the list of participants in the game:
{role_list}

Now the game has reached the voting phase. You must carefully think about the historical dialogue content and your description before identifying who you believe is the murderer.
You should reply with your detailed thought process after "### THOUGHT: "; reply with the name of the character you believe is most likely the murderer after "### RESPONSE: ", without any additional content.
Here is an example of a reply:

THOUGHT: XXX
RESPONSE: XXX

Note:
1. Your output should focus on the key elements of "Mystery Murder";
2. Your output should be in English;
3. The murderer you identify must be one of the participants listed in the game;

Your output:

Table 29: Prompt of vote

Dynamic Label Name Refinement for Few-Shot Dialogue Intent Classification

Gyutae Park[1], **Ingeol Baek**[1], **ByeongJeong Kim**[1], **Joongbo Shin**[2], **Hwanhee Lee**[1*]
[1]Department of Artificial Intelligence, Chung-Ang University, [2]LG AI Research
{pkt0401, ingeolbaek, michael97k, hwanheelee}@cau.ac.kr, jb.shin@lgresearch.ai

Abstract

Dialogue intent classification aims to identify the underlying purpose or intent of a user's input in a conversation. Current intent classification systems encounter considerable challenges, primarily due to the vast number of possible intents and the significant semantic overlap among similar intent classes. In this paper, we propose a novel approach to few-shot dialogue intent classification through in-context learning, incorporating dynamic label refinement to address these challenges. Our method retrieves relevant examples for a test input from the training set and leverages a large language model to dynamically refine intent labels based on semantic understanding, ensuring that intents are clearly distinguishable from one another. Experimental results demonstrate that our approach effectively resolves confusion between semantically similar intents, resulting in significantly enhanced performance across multiple datasets compared to baselines. We also show that our method generates more interpretable intent labels, and has a better semantic coherence in capturing underlying user intents compared to baselines. We release the code at https://github.com/lilabgyutae/Dyanmic.

1 Introduction

Dialogue intent classification identifies the underlying intent or purpose of a user's input in a conversation. It is a key component of task-oriented dialogue systems (Degand and Muller, 2020), enabling accurate understanding of user utterances and generation of appropriate responses. However, current intent classification systems face challenges, particularly in managing a large number of intent classes and resolving semantic ambiguity between similar intents (Sung et al., 2023; Cho et al., 2024; Lu et al., 2024). Recent work explores few-shot learning approaches, including retrieval-augmented methods (Milios et al., 2023; Gao et al., 2024; Abdullahi et al., 2024) and prompt-based techniques (Loukas et al., 2023; Parikh et al., 2023; Zhang et al., 2024; Rodriguez et al., 2024), which enable models to learn from limited examples per intent. While retrieval-augmented methods effectively narrow down candidate intents by retrieving examples similar to the input query, these methods also introduce a critical challenge: the retrieved examples often show significant semantic overlap across different intent categories.

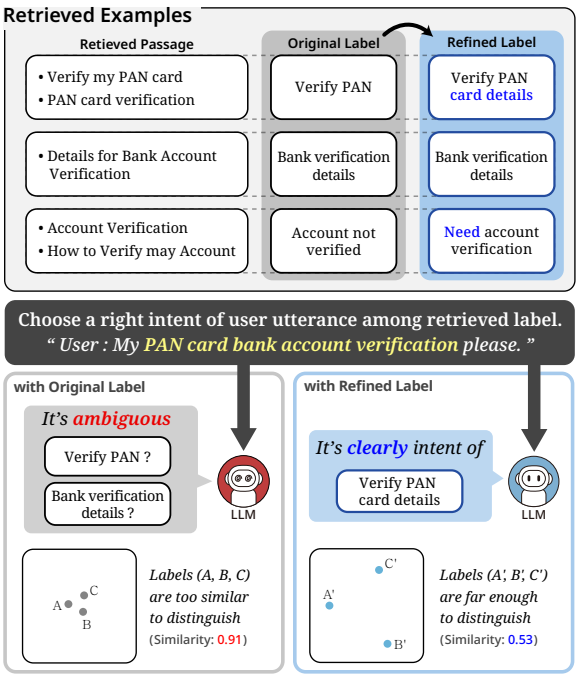


Figure 1: An example illustrating how ambiguous and similar label names can confuse the model, while refined label names enable clearer decision-making.

As shown in Figure 1, even with just three similar intents ('Verify PAN', 'Bank verification details', and 'Account not verified'), the model struggles to make accurate predictions due to their semantic similarity, as indicated by the cosine similarity score of 0.91 in the embedding space. We observe that this high semantic similarity between intent labels makes it challenging for models to

*Corresponding Author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 41–52
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

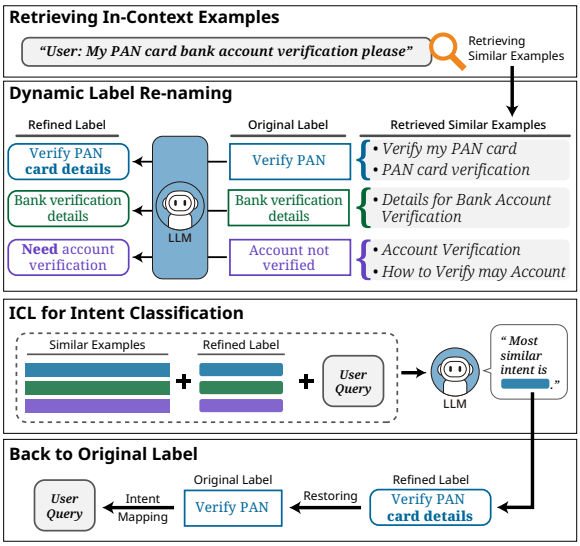


Figure 2: Overall flow of the proposed dynamic label name refinement method for intent classification.

distinguish between different intents accurately.

We find that these issues can be mitigated by refining the label names to forms that more distinctly differentiate them from other labels. As illustrated in Figure 1, by mapping the label 'Verify PAN' to a more descriptive form, such as 'Verify PAN card details', it becomes easier to differentiate it from general bank verification intents. This refinement establishes clearer semantic boundaries between intent categories, resulting in more accurate classifications.

In this paper, we present a novel approach that combines dynamic label refinement with similarity-based example selection. Our method involves retrieving semantically similar examples and dynamically refining their intent labels to create more meaningful distinctions between related intents. Through extensive experiments across various model scales and diverse datasets, we achieve significant improvements. Our analysis shows that the improvements are particularly pronounced in datasets with high semantic overlap between intents, with accuracy gains ranging from 2.07% to 7.51% across different model scales.

2 Method

Following recent work in dialogue intent classification (Milios et al., 2023; Chen et al., 2024), our approach leverages retrieval-based in-context learning (ICL) with large language models (LLMs), which has demonstrated effectiveness in tasks involving large label spaces. This approach allows models to dynamically leverage relevant few-shot examples for prediction from the training set.

We introduce a retrieval ICL method for intent classification with dynamic label re-naming, which comprises three steps: (1) retrieving semantically similar examples, (2) refining intent labels using an LLM to generate more descriptive labels, and (3) conducting the final classification with these refined examples.

2.1 Retrieving In-Context Examples

We start by retrieving relevant examples in the dataset for each input query. These retrieved examples are grouped by their original intents to provide comprehensive context for the subsequent steps.

2.2 Dynamic Label Re-naming

Our proposed dynamic label refinement process combines retrieval-based example selection with label generation guided by an LLM, as shown in Figure 2. For each of the specified intent groups, we design tailored instructions for the LLM to analyze these groups and refine the labels accordingly. (see Appendix E for full prompt.) Specifically, we design intent label refinement as a process where the model evaluates whether to retain the original label or propose an enhanced version while preserving the domain-specific semantics. During this process, the model assesses the semantic relationship between the label and its associated examples, deciding either to maintain the original intent name or to generate a more descriptive alternative. For instance, as shown in Figure 2, when analyzing examples like "*Verify my PAN card*" and "*Pan card verification*", the model recognizes that the original label '`verify_pan`' could be more descriptive and refines it to '`verify_pan_card_details`' to better capture the specific verification intent.

2.3 Dynamic vs Static Label Refinement

While a static label refinement approach might seem computationally simpler, where intent labels are refined once using all training examples collectively, our experimental validation demonstrates the superiority of dynamic refinement. Table 1 compares both approaches across representative datasets and models.

Dynamic refinement generally outperforms static refinement, though with some exceptions (e.g., static shows advantages on CUREKART and CLINC150 for smaller models). Static refinement shows two key limitations: (1) it cannot adapt to context – for example, "getting_spare_card" needs

Method	HWU64	BANKING77	CLINC150	CUREKART	POWERPLAY11	SOFMATTRESS
In-Context Learning Methods						
Llama3-8b						
Baseline	88.10	85.88	95.03	89.76	70.87	82.61
CoT	87.17 (-0.93)	85.48 (-0.40)	95.13 (+0.10)	89.76 (-0.00)	67.00 (-3.87)	81.10 (-1.51)
Static	84.10 (-4.00)	81.65 (-4.23)	87.60 (-7.43)	85.47 (-4.29)	68.61 (-2.26)	83.00 (+0.39)
Ours	**89.03** (+0.93)	**87.95** (+2.07)	**95.51** (+0.48)	**91.94** (+2.18)	**76.10** (+5.23)	**84.60** (+1.99)
Qwen2.5-7b						
Baseline	87.08	85.68	95.36	90.02	71.20	83.79
CoT	86.71 (-0.37)	87.05 (+1.37)	95.42 (+0.06)	89.98 (-0.04)	70.06 (-1.14)	84.46 (+0.67)
Ours	**88.38** (+1.30)	**87.30** (+1.62)	**95.58** (+0.22)	**90.40** (+0.38)	**74.76** (+3.56)	**87.40** (+3.61)
Qwen2.5-1.5b						
Baseline	78.90	73.31	82.29	82.78	60.84	73.12
CoT	78.72 (-0.18)	72.63 (-0.68)	81.90 (-0.39)	82.35 (-0.43)	58.83 (-2.01)	73.70 (+0.58)
Static	74.90 (-4.00)	69.18 (-4.13)	83.42 (+1.13)	**86.27** (+3.49)	55.66 (-5.18)	73.91 (+0.79)
Ours	**80.76** (+1.86)	**77.34** (+4.03)	**84.02** (+1.73)	84.10 (+1.32)	**63.10** (+2.26)	**80.63** (+7.51)
Supervised Fine-Tuning Methods						
CPFT†	87.13	87.20	94.18	-	-	-
ICDA†	87.41	**89.79**	94.84	-	-	-
QAID†	**90.42**	89.41	**95.71**	-	-	-
DF†	-	-	-	75.00	59.60	73.10
SBERT-M†	-	-	-	87.38	**64.00**	**78.78**
SBERT-P†	-	-	-	**88.05**	62.18	75.32

Table 1: Performance comparison of our dynamic label refinement approach across different models. **Baseline** represents the baseline performance using original intent labels, **CoT** shows results with chain-of-thought prompting, **Static** shows results with static label refinement, and **Ours** shows the results after applying our dynamic label refinement method. Results marked with † are from prior work. Bold indicates the best performance within each method category. Changes are computed relative to baseline.

different refinements ("additional_card_request" vs "duplicate_card_issuance") depending on context, and (2) it disrupts original semantic relationships between intents – while "cancel" naturally relates to "cancel_reservation", static refinements may obscure this connection. In contrast, dynamic refinement adapts to specific query context by refining labels based on retrieved similar examples, allowing the model to better capture semantic relationships specific to the current query context and adjust refinement strategy based on semantic similarity patterns, rather than using fixed refined labels. We also explore an alternative approach using generic identifiers instead of preserving original label names during refinement (detailed analysis in Appendix D).

2.4 ICL for Intent Classification

After obtaining the new labels for each sample, we leverage ICL with the refined labels and examples for final classification. This two-step process where the same LLM both refines the labels and makes the final classification decision helps ensure consistency between the refined semantic understanding and the ultimate intent prediction. Consequently, the overall process involves constructing a prompt that includes: 1) The retrieved examples with their refined intent labels 2) The test query requiring classification 3) Clear instructions for the model to select the most appropriate intent.

3 Experiment

3.1 Experimental Setup

Datasets We evaluate our method on two groups of datasets: DialoGLUE benchmark datasets (Mehri et al., 2020) (BANKING77, HWU64, CLINC150) and HINT3 datasets (Arora et al., 2020) (CUREKART, POWERPLAY11, SOFMATTRESS). For DialoGLUE datasets, we follow the standard 10-shot setting where each intent has only 10 examples for training. For HINT3 datasets, we exclude out-of-scope queries (`NO_NODES_DETECTED`) from evaluation as they are irrelevant to our intent classification task. Further details about datasets are provided in Appendix A.

Models We conduct experiments with three different sizes of LLMs to evaluate the effectiveness of our approach. We employ *Llama3-8b-inst.* (Dubey et al., 2024), *Qwen2.5-7b-inst.* (Yang et al., 2024), and *Qwen2.5-1.5b-inst.* as our backbone models.

Baselines We compare our approach with baseline methods across two categories: **In-context Learning Methods**: We compare against **Raw** (standard retrieval-based classification without refinement) and **Chain-of-Thought (CoT)** (Wei et al., 2023) (structured reasoning approach). Our method (**Ours**) applies dynamic label refinement to the retrieved examples before classification. **Supervised Fine-Tuning Methods**: We compare with state-of-the-art SFT-based approaches including **CPFT** (Zhang et al., 2021), **ICDA** (Lin et al., 2023), **QAID** (Yehudai et al., 2023) for DialoGLUE datasets, and **DF**, **SBERT-M**, **SBERT-P** (Arora et al., 2020) for HINT3 datasets.

Implementation Details For retrieving semantically similar examples, we use a pre-trained SentenceTransformer model (Reimers, 2019). For each input query, we retrieve the top-20 most similar examples. Following (Milios et al., 2023), we order examples from least to most (Zhou et al., 2022) similar in the prompt, which demonstrated higher accuracy across our datasets. We use this retrieval-based in-context learning setup as our baseline, where the LLM directly performs classification using the original intent labels. While a static label refinement might seem simpler, we opt for dynamic refinement as it enables context-specific label adjustments based on each test query and its retrieved examples. We confirm this choice through experimental validation in Section 2.3.

3.2 Main Results

Table 1 presents a comprehensive analysis of our dynamic label refinement approach across different experimental setups. We structure the investigation around several critical dimensions to clearly demonstrate the effectiveness and implications of our method.

Semantic Disambiguation through Label Refinement We first confirm that while both Raw and CoT approaches show reasonable performance, they often struggle with semantically ambiguous intents. For example, CoT misclassifies "*Please keep delivery service to the pin code 7021*" as '`modify_address`' instead of '`check_pincode`', and "*Hey I didn't receive the ordered product its incomplete*" as '`refunds_returns_replacements`' instead of '`delay_in_parcel`'. These examples show how structured reasoning often struggles when similar intents have overlapping semantics. Our refinement process tackles this issue by analyzing the semantic relationships between labels and examples, leading to consistent performance improvements over both Raw and CoT baselines across all datasets as shown in Table 1. The improvements are particularly notable in datasets with complex domain-specific terminology, such as BANKING77, where the model effectively leverages existing semantic information in the labels.

Performance Across Model Scales Our experiments with different model sizes reveal several interesting patterns about the scalability of our approach. The larger models (Llama3-8b-inst. and Qwen2.5-7b-inst.) demonstrate robust baseline performance with moderate improvements across all datasets. Notably, even the smaller Qwen2.5-1.5b-inst. Model achieves significant improvements, particularly on domain-specific datasets (SOFMATTRESS, BANKING77), suggesting that our approach effectively enhances performance regardless of model scale. These results demonstrate that our proposed method effectively enhances intent classification performance across various datasets and model architectures. Additional experiments with larger models showing similar trends can be found in Appendix I.

3.3 Analysis

Semantic Similarity Analysis To validate our hypothesis about semantic relationships between intent labels, we conduct an embedding-based similarity analysis comparing original and refined intent labels across all datasets. Specifically, we leverage the last hidden layer representations from our LLMs to capture the semantic characteristics of each intent label. For each intent, we extract the final hidden state representation and computed pairwise cosine similarities between these representations within each label set.

We observe that refined labels consistently achieve lower average pairwise similarities than original labels across all datasets and model scales, as shown in Table 2. With Llama3-8b-inst., the average similarity drops from 0.86 for original intent labels to 0.74 for refined labels. Qwen2.5-7b-

Model	Original	Refined
Llama3-8b-inst.	0.86	**0.74**
Qwen2.5-7b-inst.	0.83	**0.80**
Qwen2.5-1.5b-inst.	0.95	**0.91**

Table 2: Average pairwise semantic similarity between original and refined intent labels across various model scales.

Dataset	Q2.5-7B		Q2.5-1.5B	
	Q2.5-7B	Q2.5-1.5B	Q2.5-1.5B	Q2.5-7B
HWU64	**88.38**	81.22	80.76	**88.01**
BANKING77	**87.30**	80.81	77.34	**85.95**
CLINC150	**95.58**	87.31	84.02	**95.22**
CUREKART	**90.40**	82.78	84.10	**88.90**
POWERPLAY11	**74.76**	64.10	63.10	**74.43**
SOFMATTRESS	**87.40**	78.30	80.63	**85.08**

Table 3: Performance comparison across different model combinations. The top row means the model used for re-naming, and the second row denotes the models used for classification. Models used: Qwen2.5-7B-inst. (Q2.5-7B) and Qwen2.5-1.5B-inst. (Q2.5-1.5B).

inst. and Qwen2.5-1.5b-inst. follow a similar trend. This decrease in semantic overlap directly correlates with improved classification performance. For instance, on the BANKING77 dataset, Llama3-8b-inst. achieves a 2.07% accuracy improvement along with a 0.1 reduction in label similarity. The performance gains become especially noticeable when the model creates more semantically distinct labels, indicating that reducing label overlap enables clearer distinctions between different intents.

Model Combination To validate the effectiveness of labeling refinement itself of each model, we employ two separate LLMs for intent refinement and classification tasks. We experiment with various combinations of Qwen2.5-7B-inst. and Qwen2.5-1.5B-inst. As in Table 3, using a larger model for refinement followed by a smaller model for classification yields the best performance. For example, using Qwen2.5-7B-inst. for refinement and Qwen2.5-1.5B-inst. for answering achieves notable improvements: +3.31% on CLINC150 and +4.10% on POWERPLAY11 compared to single-model baselines.

Interestingly, even reverse combinations (small model refinement + large model classification) show improvements over the non-refinement baseline, though to a lesser extent. For instance, using Qwen2.5-1.5B-inst. for refinement and Qwen2.5-7B-inst. for answering still achieves improvements on the datasets. This suggests that our label refinement approach is robust across different model scales and configurations. A detailed analysis of performance across different model combinations is provided in Appendix H.

4 Conclusion

We propose a dynamic label refinement method for few-shot dialogue intent classification that mitigates the issues of significant semantic overlap between intent labels. Using the retrieved examples, we refine labels via LLMs to create more semantically distinct intent categories. Experimental results demonstrate that our method consistently improves performance across multiple datasets for various models. We also confirm that our method reduces semantic similarities between intent labels, creating more distinct and interpretable categories.

Limitations

While our approach demonstrates significant improvements in intent classification performance, it requires additional computational overhead compared to traditional methods. The need to run the model once for label refinement and once for classification - increases the computational cost per query. However, we believe this trade-off is justified by the substantial improvements in classification accuracy, particularly for semantically ambiguous intents.

Acknowledgement

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2021II211341, Artificial Intelligence Graduate School Program (Chung-Ang University)] and the Chung-Ang University Graduate Research Scholarship in 2023 and was improved by the helpful input and collaboration of researchers from LG AI Research.

References

Tassallah Abdullahi, Ritambhara Singh, and Carsten Eickhoff. 2024. Retrieval augmented zero-shot text classification. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '24, page 195–203. ACM.

Gaurav Arora, Chirag Jain, Manas Chaturvedi, and Krupal Modi. 2020. HINT3: Raising the bar for intent detection in the wild. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*,

pages 100–105, Online. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Huiyao Chen, Yu Zhao, Zulong Chen, Mengjia Wang, Liangyue Li, Meishan Zhang, and Min Zhang. 2024. Retrieval-style in-context learning for few-shot hierarchical text classification. *Preprint*, arXiv:2406.17534.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *Preprint*, arXiv:1902.10909.

Jaejin Cho, Rakshith Sharma Srinivasa, Ching-Hua Lee, Yashas Malur Saidutta, Chouchang Yang, Yilin Shen, and Hongxia Jin. 2024. Zero-shot intent classification using a semantic similarity aware contrastive loss and large language model. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10776–10780. IEEE.

Liesbeth Degand and Philippe Muller. 2020. Introduction to the special issue on dialogue and dialogue systems. *Traitement Automatique des Langues*, 61(3):7–15.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Lingyu Gao, Aditi Chaudhary, Krishna Srinivasan, Kazuma Hashimoto, Karthik Raman, and Michael Bendersky. 2024. Ambiguity-aware in-context learning with large language models. *Preprint*, arXiv:2309.07900.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Yen-Ting Lin, Alexandros Papangelis, Seokhwan Kim, Sungjin Lee, Devamanyu Hazarika, Mahdi Namazifar, Di Jin, Yang Liu, and Dilek Hakkani-Tur. 2023. Selective in-context data augmentation for intent detection using pointwise V-information. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1463–1476, Dubrovnik, Croatia. Association for Computational Linguistics.

Lefteris Loukas, Ilias Stogiannidis, Prodromos Malakasiotis, and Stavros Vassos. 2023. Breaking the bank with ChatGPT: Few-shot text classification for finance. In *Proceedings of the Fifth Workshop on Financial Technology and Natural Language Processing and the Second Multimodal AI For Financial Forecasting*, pages 74–80, Macao. -.

Zhenyi Lu, Jie Tian, Wei Wei, Xiaoye Qu, Yu Cheng, Wenfeng xie, and Dangyang Chen. 2024. Mitigating boundary ambiguity and inherent bias for text classification in the era of large language models. *Preprint*, arXiv:2406.07001.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tur. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *Preprint*, arXiv:2009.13570.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. 2023. In-context learning for text classification with many labels. *Preprint*, arXiv:2309.10954.

Soham Parikh, Quaizar Vohra, Prashil Tumbade, and Mitul Tiwari. 2023. Exploring zero and few-shot techniques for intent classification. *Preprint*, arXiv:2305.07157.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Juan A. Rodriguez, Nicholas Botzer, David Vazquez, Christopher Pal, Marco Pedersoli, and Issam Laradji. 2024. Intentgpt: Few-shot intent discovery with large language models. *Preprint*, arXiv:2411.10670.

Mujeen Sung, James Gung, Elman Mansimov, Nikolaos Pappas, Raphael Shu, Salvatore Romeo, Yi Zhang, and Vittorio Castelli. 2023. Pre-training intent-aware encoders for zero- and few-shot intent classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10433–10442, Singapore. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Asaf Yehudai, Matan Vetzler, Yosi Mass, Koren Lazar, Doron Cohen, and Boaz Carmeli. 2023. Qaid: Question answering inspired few-shot intent detection. *Preprint*, arXiv:2303.01593.

Haode Zhang, Haowen Liang, Liming Zhan, Albert Y. S. Lam, and Xiao-Ming Wu. 2024. Revisit few-shot intent classification with plms: Direct fine-tuning vs. continual pre-training. *Preprint*, arXiv:2306.05278.

Jianguo Zhang, Trung Bui, Seunghyun Yoon, Xiang Chen, Zhiwei Liu, Congying Xia, Quan Hung Tran, Walter Chang, and Philip Yu. 2021. Few-shot intent detection via contrastive pre-training and fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1906–1912, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Dataset Details

We evaluate our approach using two different groups of datasets. The first group consists of DialoGLUE benchmark datasets (**BANKING77**: 77 intents focused on banking domain (Casanueva et al., 2020), **HWU64**: 64 intents spanning across 21 domains (Casanueva et al., 2020), **CLINC150**: 150 intents covering 10 domains (Larson et al., 2019)), which are widely used for evaluating task-oriented dialogue systems. For these datasets, we follow the standard 10-shot setting where each intent class has only 10 examples for training, reflecting real-world scenarios where collecting large amounts of labeled data is challenging.

The second group includes HINT3 datasets (Arora et al., 2020) (**CUREKART**: 28 intents in fitness supplements retail domain, **POWERPLAY11**: 59 intents in online gaming domain, **SOFMATTRESS**: 21 intents in mattress products retail domain), which contain real user queries from live chatbots. We exclude the `NO_NODES_DETECTED` label from the test set as it represents out-of-scope queries irrelevant to our task.

B Impact of the Number of Retrieved Examples

We conduct additional experiments to analyze the impact of the number of retrieved examples on model performance. Figure 3 shows the performance comparison between different model configurations across varying numbers of retrieved examples (10, 20, 30, and 40) on BANKING77 and POWERPLAY11 datasets.

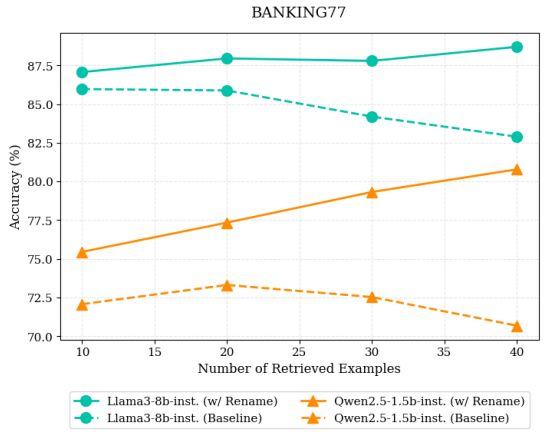
From these results, we observe several key patterns:

• The performance gap between the baseline and refined versions tends to be more pronounced with larger numbers of examples, particularly in BANKING77.

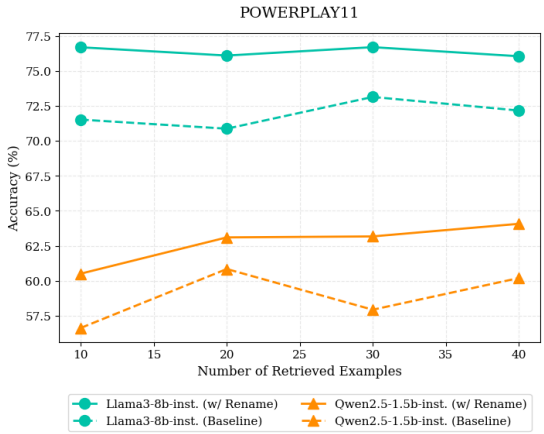
• The larger model (Llama3-8b-inst.) shows more stable performance across different example counts, while the smaller model (Qwen2.5-1.5b-inst.) shows greater variance in performance.

• POWERPLAY11 shows relatively consistent improvement patterns across different example counts, suggesting that the benefits of label refinement are robust across different dataset characteristics.

These findings suggest that our label refinement approach is effective across different numbers of retrieved examples.



BANKING77



POWERPLAY11

Figure 3: Performance comparison with different numbers of retrieved examples. Solid lines represent performance with label refinement, while dashed lines represent baseline performance without refinement.

C Dataset-wise Similarity Analysis

To provide a detailed view of semantic similarity patterns, we analyze the pairwise similarities between intent labels for each dataset. Table 5shows the average pairwise similarity scores for both original and refined intent labels across different datasets and models.

As shown in Table 5, the reduction in semantic similarity is consistent across all datasets for both models, though the magnitude of reduction varies. For Llama3-8b-inst., POWERPLAY11 shows the largest reduction in similarity (0.156), while CUREKART shows the smallest (0.079). Similarly, for Qwen2.5-1.5b-inst., CLINC150 shows a notable reduction (0.045) while BANKING77 shows a relatively smaller

Dataset	Recall@20	Avg. Intents
HWU64	97.77	7.54
BANKING77	98.93	7.04
CLINC150	99.33	6.31
CUREKART	98.91	4.31
SOFMATTRESS	98.81	5.86
POWERPLAY11	96.44	6.93

Table 4: Retrieval analysis with 20 examples

Dataset	Llama3-8b-inst.		Qwen2.5-7b-inst.		Qwen2.5-1.5b-inst.	
	Original	Refined	Original	Refined	Original	Refined
HWU64	0.835	0.721	0.772	0.760	0.948	0.910
BANKING77	0.889	0.786	0.852	0.838	0.955	0.938
CLINC150	0.834	0.705	0.780	0.764	0.946	0.901
CUREKART	0.881	0.802	0.800	0.833	0.956	0.925
SOFMATTRESS	0.849	0.761	0.752	0.748	0.943	0.911
POWERPLAY11	0.844	0.688	0.838	0.812	0.945	0.903
Mean	0.856	0.744	0.799	0.793	0.949	0.915

Table 5: Detailed semantic similarity analysis across datasets and models. Values represent the average pairwise cosine similarity between all intent labels within each dataset. Lower values indicate more semantically distinct intent categories.

change (0.017). These variations might reflect differences in the initial intent label structures across datasets and the models' ability to refine them effectively.

D Semantic vs Generic Label Refinement

While our main experiments focus on semantic-aware refinement that preserves domain context, we also explore an alternative approach using generic identifiers (w/o ori). In this setting, all original intent labels are first replaced with generic identifiers (e.g., Intent_1, Intent_2) before refinement. During refinement, the model generates new labels without any influence from the original intent names.

The effectiveness of each approach varies with model size. Larger models (Llama3-8b-inst. and Qwen2.5-7b-inst.) generally perform better with original name preservation (w ori), particularly in BANKING77 where domain-specific terminology provides valuable semantic context. However, smaller models (Qwen2.5-1.5b-inst.) often show improved performance with generic identifiers (w/o ori), suggesting that they may benefit from the simplified label space. This performance pattern indicates that the choice between preserving or abstracting intent names should consider the model's capacity to leverage domain-specific terminology.

Complete results comparing both approaches across all datasets and models are shown in Table 6.

E Full Prompt Template

E.1 Re-naming Prompt Template

The re-naming prompt takes groups of examples organized by their original intent labels and analyzes their semantic meaning. Based on this analysis, it either keeps the original label if it accurately represents the examples, or generates a new more descriptive label while maintaining consistent formatting rules. The prompt enforces lowercase letters and underscores in label names to ensure standardization.

E.2 Classification Prompt Template

The classification prompt presents example pairs of text queries and their corresponding refined intents to establish the task context. It instructs the model to determine the most likely intent for a new query based solely on these examples. The prompt explicitly requires only the intent name as output, without any additional explanation or text, to ensure consistent and clean predictions.

E.3 Chain-of-Thought Prompt Template

The Chain-of-Thought prompt extends the basic classification prompt by requiring the model to explain its reasoning process. For each query, the model must analyze the key elements, provide reasoning, and then determine the intent. This structured approach aims to help the model make more informed decisions by breaking down the classification process into steps.

F Related Work

F.1 Dialogue Intent Classification

Dialogue intent classification aims to identify users' intentions from natural language utterances. Traditional approaches relied on supervised learning with large labeled datasets (Chen et al., 2019; Larson et al., 2019). The advent of large language models (LLMs) transforms this landscape, enabling effective few-shot learning approaches where only limited labeled data is available (Brown et al., 2020). This shift has been particularly significant as LLMs demonstrate strong few-shot capabilities through in-context learning, reducing the need for extensive labeled datasets (Loukas et al., 2023; Parikh et al., 2023; Chen et al., 2024).

Dataset	Llama3-8b-inst			Qwen2.5-7b-inst			Qwen2.5 -1.5b-inst		
	Raw	Refined w/o ori	Refined w ori	Raw	Refined w/o ori	Refined w ori	Raw	Refined w/o ori	Refined w ori
HWU64	88.10	87.17 (-0.93)	**89.03** (+0.93)	87.08	85.97 (-1.11)	**88.38** (+1.30)	78.90	79.27 (+0.37)	**80.76** (+1.86)
BANKING77	85.88	87.27 (+1.39)	**87.95** (+2.07)	85.68	86.33 (+0.65)	**87.30** (+1.62)	73.31	**79.12** (+5.81)	77.34 (+4.03)
CLINC150	95.03	93.73 (-1.30)	**95.51** (+0.48)	95.36	93.73 (-1.63)	**95.58** (+0.22)	82.29	**86.38** (+4.09)	84.02 (+1.73)
CUREKART	89.76	**91.94** (+2.18)	**91.94** (+2.18)	90.02	**90.40** (+0.38)	**90.40** (+0.38)	82.78	**86.27** (+3.49)	84.10 (+1.32)
POWERPLAY11	70.87	76.05 (+5.18)	**76.10** (+5.23)	71.20	74.43 (+3.23)	**74.76** (+3.56)	60.84	**66.99** (+6.15)	63.10 (+2.26)
SOFMATTRESS	82.61	83.40 (+0.79)	**85.40** (+2.79)	83.79	82.60 (-1.19)	**87.40** (+3.61)	73.12	79.44 (+6.32)	**80.63** (+7.51)

Table 6: Complete performance comparison including refinement without original name preservation (Refined w/o ori). Results show that while both refinement approaches generally improve over the baseline (Raw), preserving original names during refinement (Refined w ori) tends to yield better or comparable results.

F.2 Retrieval-based In-Context Learning

Retrieval-based approaches have emerged as a powerful paradigm for improving few-shot learning performance. Key developments in this area include: (Milios et al., 2023) propose effective retrieval strategies for in-context learning with many labels, demonstrating significant performance improvements through careful example selection. However, while this approach successfully retrieves semantically similar examples for classification, it introduces new challenges when dealing with intent labels that have high semantic overlap. (Lu et al., 2024) This ambiguity between similar intents creates unnecessary complexity in the classification task, particularly when multiple intents share similar contextual meanings but require different downstream processing. Our work addresses this challenge by introducing a dynamic label refinement approach that helps distinguish between semantically similar intents while maintaining the benefits of retrieval-based example selection.

G Case Study

G.1 Label Refinement Pattern Analysis

Our analysis revealed interesting patterns in how the model refines intent labels, particularly highlighting some suboptimal refinement behaviors:

G.1.1 Verbatim Query-to-Intent Conversion

We observed cases where the model simply converted user queries directly into intent labels:

- Original_text: "I want to return my mattress"
- Refined_intent: `i_want_to_return_my_mattress`

This pattern indicates a potential limitation in our refinement approach where the model sometimes fails to abstract the core intent, instead creating overly specific labels that mirror the input text.

G.1.2 Overly Descriptive Intent Labels

Another pattern emerged where the model generated unnecessarily verbose intent labels:

- Original_intent: `size_customization`
- Refined_intent: `how_can_i_order_a_custom_sized_mattress`

These findings highlight the need for more sophisticated label refinement strategies that maintain a balance between descriptiveness and practical utility.

H Performance comparison across different model combinations.

Based on Table 3, we conduct additional experiments employing two separate large language models (LLMs) for intent segmentation and classification tasks in order to verify the effectiveness of label segmentation for each model. In particular, for the two models used in the experiment, we report the accuracy of both the model that generates the intents and the model that generates the responses. Consistent with the results shown in Table 3, we find that using a larger model for response generation is effective. Furthermore, we observe that label re-naming, even when performed using labels derived from a smaller model, still yields strong performance.

I Additional Experiments with Large Models

To further validate the effectiveness of our approach across different model scales, we conducted

Examples by intent:
Intent 1: account_not_verified
- How to Verify my Account?
- Account Verification
- Need to Verify my account

Intent 2: delete_pan_card
- Pan card remove
- Delete PAN card
- I want to delete my pan card

Intent 3: bank_verification_details
...
Intent 4: pan_verification_failed
...
Rules for intent mapping:
1. If the current intent name accurately represents its examples, map it to itself
2. If the intent name needs improvement, create a new descriptive name that better represents the examples
3. For new names:
- Use lowercase letters only
- Use underscores between words

INTENT MAPPINGS:
account_not_verified ->
delete_pan_card ->
...
pan_verification_failed ->

Table 7: Re-naming prompt template.

additional experiments with larger models including ChatGPT3.5-turbo-0125 and Llama3-70b-inst.

Results on six benchmark datasets are shown in Table 11 demonstrate that our label refinement approach consistently improves performance even with larger models. Key observations include:

- Both models show consistent improvements across all datasets, with gains ranging from +0.20% to +2.33%
- ChatGPT3.5-turbo-0125 shows particularly strong improvements on CUREKART (+2.33%) and POWERPLAY11 (+1.13%)
- Llama3-70b-inst. achieves notable gains on CUREKART (+1.99%) and SOFMATTRESS (+1.61%)

These results further validate that our label refinement approach is effective across different model scales.

You are an AI assistant specialized in intent classification. Your task is to determine
the single most likely intent of a given query based on the examples provided.
Provide only the name of the most probable intent, without any additional text or explanation.

...

Text: "Details for Bank Account Verification"
Intent: bank_verification_details

Text: "Getting error while verifying PAN Card"
Intent: pan_verification_failure

Text: "My PAN card needs to be verified"
Intent: verify_pan_card_details

Query: "My PAN card bank account verification please"
The top 1 most likely intent is:

Table 8: Classification prompt template.

You are an AI assistant specialized in intent classification. Your task is to determine
the single most likely intent of a given query based on the examples provided.
For each query:
1. Analyze the key elements and meaning
2. Provide an explanation of your reasoning
3. Extract the most likely intent

Text: "Details for Bank Account Verification"
Intent: bank_verification_details

Text: "Getting error while verifying PAN Card"
Intent: pan_verification_failure

Text: "My PAN card needs to be verified"
Intent: verify_pan_card_details

Query: "My PAN card bank account verification please"
Provide your explanation and intent:

Table 9: Chain-of-Thought prompt template.

Dataset	L3-8B		Q2.5-1.5B	
	L3-8B	Q2.5-1.5B	Q2.5-1.5B	L3-8B
HWU64	**89.03**	80.95	80.76	**88.48**
BANKING77	**87.95**	80.09	77.34	**88.47**
CLINC150	**95.51**	86.82	84.02	**94.79**
CUREKART	**91.94**	84.96	84.10	**91.07**
POWERPLAY11	**76.10**	65.69	63.10	**74.11**
SOFMATTRESS	**85.40**	79.44	80.63	**85.38**

Table 10: Performance comparison across different model combinations. The top row shows the model used for re-naming, while the models listed in the second row are used for answering (classification).

ChatGPT3.5-turbo-0125		
Dataset	Baseline	Re-naming
HWU64	89.48	89.70 (+0.22)
BANKING77	88.78	89.34 (+0.56)
CLINC150	96.35	96.67 (+0.32)
CUREKART	90.04	92.37 (+2.33)
POWERPLAY11	75.08	76.21 (+1.13)
SOFMATTRESS	84.18	85.27 (+1.09)
Llama3-70b-inst.		
Dataset	Baseline	Re-naming
HWU64	89.31	89.56 (+0.25)
BANKING77	88.31	88.62 (+0.31)
CLINC150	96.32	96.52 (+0.20)
CUREKART	90.04	92.03 (+1.99)
POWERPLAY11	73.46	74.80 (+1.34)
SOFMATTRESS	83.79	85.40 (+1.61)

Table 11: Performance comparison with large language models. Numbers in parentheses show absolute improvement over baseline.

Rethinking KenLM: Good and Bad Model Ensembles for Efficient Text Quality Filtering in Large Web Corpora

Yungi Kim[1*], Hyunsoo Ha[2*], Sukyung Lee[2], Jihoo Kim[2], Seonghoon Yang[2], Chanjun Park[3†]

[1]Liner, [2]Upstage AI, [3]Korea University
eddie@linercorp.com
{hyunsooha, sukyung, jerry, hoonyang}@upstage.ai
bcj1210@korea.ac.kr

Abstract

With the increasing demand for substantial amounts of high-quality data to train large language models (LLMs), efficiently filtering large web corpora has become a critical challenge. For this purpose, KenLM, a lightweight n-gram-based language model that operates on CPUs, is widely used. However, the traditional method of training KenLM utilizes only high-quality data and, consequently, does not *explicitly* learn the linguistic patterns of low-quality data. To address this issue, we propose an ensemble approach that leverages two contrasting KenLMs: (i) *Good KenLM*, trained on high-quality data; and (ii) *Bad KenLM*, trained on low-quality data. Experimental results demonstrate that our approach significantly reduces noisy content while preserving high-quality content compared to the traditional KenLM training method. This indicates that our method can be a practical solution with minimal computational overhead for resource-constrained environments.

1 Introduction

The advancement of large language models (LLMs) has accelerated as the *'scaling law'* (Kaplan et al., 2020), which states that the performance of LLMs directly correlates with data size, has been studied. Moreover, recent studies (Penedo et al., 2023; Gunasekar et al., 2023; Li et al., 2024; Penedo et al., 2024; Dubey et al., 2024) have shown that the performance of LLMs is largely determined by the quality of the training corpus. In other words, a vast amount of high-quality training corpus is necessary to enhance the performance of LLMs.

However, large web corpora often contain substantial amounts of low-quality data, making them difficult to use directly for training. In response to this challenge, various methods (Wettig et al., 2024; Kong et al., 2024) are employed to filter out low-quality data and select high-quality data. These methods typically require GPU resources, which makes them impractical, especially when processing data that exceeds trillions of tokens.

To *efficiently* filter large datasets, the most widely used method is KenLM (Heafield, 2011), a lightweight n-gram-based model that operates on *CPUs*. In many studies (Wenzek et al., 2019; Computer, 2023; Nguyen et al., 2023; Laurençon et al., 2024), KenLM, trained on the high-quality Wikipedia dataset, is commonly used. It measures perplexity (PPL) to identify low-quality content. Note that higher PPL scores indicate lower-quality or out-of-domain text, while lower PPL scores suggest that the text closely resembles the linguistic patterns of the high-quality data used to train KenLM. Low-quality data with high PPL scores are then filtered out.

We argue that the traditional KenLM does not *explicitly* learn the linguistic patterns of low-quality data. Thus, while it assigns low PPL scores to data with high-quality linguistic patterns, it does not *consistently* assign high PPL scores to data with low-quality linguistic patterns. To address this issue, we propose an ensemble approach that utilizes the following two contrasting KenLMs: (i) *Good KenLM*, trained on high-quality data; and (ii) *Bad KenLM*, trained on noisy, low-quality data such as spam emails, hate speech, and informal social media text. Our empirical results show that this approach can be a practical solution with minimal computational overhead for resource-constrained environments, significantly reducing noisy content and preserving high-quality content compared to the traditional KenLM training method.

2 Related Work

As the demand for a vast amount of high-quality training corpus grows, it has become essential to *effectively* and *efficiently* filter large amounts of web corpus. Among various filtering methods,

*Equal Contribution † Corresponding Author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 53–58
July 27 - August 1, 2025

this paper focuses on model-based quality filtering, which can be broadly divided into the following two categories: (i) perplexity-based filtering; and (ii) classifier-based filtering.

Perplexity-based filtering. Numerous studies (Wenzek et al., 2019; Computer, 2023; Nguyen et al., 2023; Wei et al., 2023; Paster et al., 2023; Laurençon et al., 2024) use the perplexity (PPL) scores of KenLM (Heafield, 2011), an n-gram-based language model, to *efficiently* filter out low-quality data due to its lightweight architecture. It can operate on *CPUs*, making it a cost-efficient solution for handling large-scale text data. Despite its efficiency, there have been few efforts to improve its performance. Meanwhile, The Pile (Gao et al., 2020) used the perplexity of GPT-2 (Radford et al., 2019) and GPT-3 (Brown, 2020) to evaluate the quality of the dataset.

Classifier-based filtering. FastText (Joulin et al., 2016) is widely used to distinguish the quality of data (Computer, 2023; Wei et al., 2023; Li et al., 2024). Similar to KenLM, FastText is also an efficient model that operates on CPUs. However, as detailed in Section 4, KenLM demonstrated superior performance compared to FastText when both were trained on the same dataset. Furthermore, recent research (Gunasekar et al., 2023; Li et al., 2024; Penedo et al., 2024) has focused on fine-tuning pre-trained embedding models to serve as classifiers for quality filtering. Especially, Fineweb demonstrated that training relatively small-sized LLMs (1.82 billion parameters) on data filtered by a trained classifier (resulting in 350 billion tokens), rather than on unfiltered data, led to performance improvements across various benchmarks. However, these methods are impractical for processing large web corpora due to their high computational costs, which necessitate significant GPU resources.

3 Proposed Method

In this paper, we aim to reduce noisy data while preserving high-quality data in a computationally efficient manner. To this end, we propose an ensemble approach using two contrasting KenLMs: (i) Good KenLM and (ii) Bad KenLM.

Good KenLM. The objective of Good KenLM is to assign low perplexity (PPL) scores to well-structured, high-quality text. Many previous studies (Wenzek et al., 2019; Computer, 2023; Nguyen et al., 2023; Laurençon et al., 2024) have used a high-quality Wikipedia dataset for training, denoted as Wiki KenLM in this paper. However, with recent advancements in LLMs, several high-quality datasets (Soldaini et al., 2024; Penedo et al., 2024; Li et al., 2024) have emerged. In our experiments, as shown in Section 4, we found that the combination of S2ORC (Lo et al., 2020) and Textbooks-are-all-you-need-lite (SciPhi, 2023) as training data was more effective than utilizing Wikipedia. Thus, in this paper, we designate the KenLM trained on this combination of data as Good KenLM.

Bad KenLM. The rationale behind employing Bad KenLM alongside Good KenLM is that Good KenLM fails to detect unwanted content (*e.g.*, spam, advertising, and informal communication), which are generally considered poor for training LLMs, as it has not been *explicitly* trained on these types of content. For instance, if low-quality content shares superficial linguistic patterns with high-quality text, it may still score reasonably well under Good KenLM. Therefore, to detect a wider range of undesirable content, Bad KenLM is designed to assign low PPL scores to such content. To achieve this, we trained Bad KenLM using noisy, low-quality datasets, including hate speech, spam emails, and informal social media content. To the best of our knowledge, this is the first study to employ KenLM trained on noisy, low-quality datasets.

Ensemble. To leverage the complementary strengths of two contrasting KenLMs, we ensemble the models by integrating the PPL scores assigned by each. We perform Z-score standardization to align the scales of the two PPL scores assigned by each model, as they are trained on different datasets and therefore exhibit different distributions of PPL scores. Then, we compute the ensembled PPL score $P_{ens}(x_i)$, as follows:

$$P_{\text{ens}}(x_i) = \alpha \left(\frac{P_{\text{good}}(x_i) - \mu_{\text{good}}}{\sigma_{\text{good}}} \right) - (1-\alpha) \left(\frac{P_{\text{bad}}(x_i) - \mu_{\text{bad}}}{\sigma_{\text{bad}}} \right), \quad (1)$$

where $x_i \in \mathcal{X}$ denotes the i-th text data, $\mathcal{X}$ represents datasets, $P_{\text{good}}(x_i)$ (resp. $P_{\text{bad}}(x_i)$) indicates PPL score from Good (resp. Bad) KenLM for x_i, μ_{good} (resp. μ_{bad}) is the mean of the PPL scores from Good (resp. Bad) KenLM, σ_{good} (resp. σ_{bad}) is the standard deviation of the PPL scores from Good (resp. Bad) KenLM, and α denotes a parameter that balances the two PPL scores. Note that the coefficient for the term associated with Bad

KenLM is negative. This is because, in contrast to Good KenLM, which assigns low PPL scores to high-quality data, Bad KenLM assigns low PPL scores to low-quality data. Consequently, data with low ensembled PPL scores—obtained by appropriately subtracting two PPL scores—closely resemble the linguistic patterns of high-quality data and are distinctly separated from low-quality content.

4 Experiments

We designed our experiments to answer the following key research questions (RQs):

- **RQ1**: Does our ensemble approach outperform existing models in removing noisy content while preserving high-quality content?
- **RQ2**: Which data sources are effective for training the Bad KenLM?
- **RQ3**: How sensitive is the performance of our ensemble approach to hyperparameter α?
- **RQ4**: How much additional computational overhead does our ensemble approach introduce compared to a single KenLM?
- **RQ5**: What types of data does our ensemble approach effectively filter out?

4.1 Experimental Settings

Dataset and model details. As mentioned in Section 3, we randomly selected subsets of 300,000 samples each from S2ORC (Lo et al., 2020) and Textbooks-are-all-you-need-lite (SciPhi, 2023) as training data for Good KenLM. For the training data of Bad KenLM, we collected datasets that is likely to hinder the training of LLMs. Specifically, we used 1,000,000 pieces of social network service (SNS) data (Twitter) (mjw, 2022; fschatt, 2023; lcama, 2022; StephanAkkerman, 2023) and 776,142 pieces of spam message data (Metsis et al., 2006; thehamkercat, 2024; alissonpadua, 2024). During the training of both models, we configured the n-gram size to 6 and the vocabulary size to $65,536$. Also, we set the hyperparameter α to 0.7.

Evaluation details. To evaluate the effectiveness of our ensemble approach, we measured perplexity (PPL) scores for the CC-MAIN-2024-10 dump (211 million samples) from Fineweb-edu (Penedo et al., 2024). Following Wenzek et al. (2019); Computer (2023), we then filtered the data based on the 30th and 60th percentiles of PPL scores. Subsequently, we measured the proportion of data with an *educational score* of 2.5 or higher that was included. In other words, we treated data with an educational score of 2.5 or higher as the ground truth and measured the recall value. Note that the educational scores are annotated using extensive GPU resources, and it has been demonstrated that training LLMs with data possessing high educational scores leads to performance improvements.

Models	Recall@30	Recall@60	Average Recall
Wiki KenLM	0.5530	0.8513	0.7022
Good KenLM	0.7059	0.9195	0.8127
Bad KenLM	0.3403	0.7031	0.5217
FastText(Wiki, Bad)	0.6453	0.8878	0.7665
FastText(Good, Bad)	0.7462	0.9412	0.8437
Ens(Good, Bad)	**0.8190**	**0.9647**	**0.8919**
Ens(Good, Wiki)	0.6312	0.8898	0.7605

Table 1: Performance comparison of our approach with existing models, and an ablation study on our design choices.

4.2 Main Results

We highlight the best results in bold and the second-best results with an underline in the tables.

RQ1: Comparison of existing models. As shown in Table 1, our Good KenLM significantly outperformed the widely used Wiki KenLM. Although Bad KenLM alone showed poor performance, our strategy of ensembling it with Good KenLM outperformed even FastText trained on the same data, improving Recall@30 and Recall@60 by 9.76% and 2.50%, respectively.

Moreover, to validate the effectiveness of Bad KenLM within our ensemble framework, we conducted a comparative experiment where Good KenLM and Wiki KenLM were ensembled in place of Bad KenLM, denoted as *Ens(Good, Wiki)*. The performance of *Ens(Good, Wiki)* was lower than that of Good KenLM alone. This is likely due to the relatively lower quality of the Wikipedia dataset compared to the training data used for Good KenLM, which negatively impacts its overall performance. This result also highlight the importance of incorporating Bad KenLM into the ensemble, as it successfully identifies undesirable content that Good KenLM may overlook.

RQ2: Impact of data sources on training Bad KenLM. The training dataset for Bad KenLM is diverse, including SNS, spam mail, and toxic datasets (Davidson et al., 2017; de Gibert et al., 2018; Kennedy et al., 2020; Mathew et al., 2021; Vidgen et al., 2021; Pavlopoulos et al., 2022) containing hate speech and profanity. We conducted

Training Dataset of Bad KenLM	Metrics		
	Recall@30	Recall@60	Average Recall
Spam	0.8059	0.9576	0.8818
Twitter	0.8131	**0.9651**	0.8891
Toxic	0.7320	0.9402	0.8361
Spam + Twitter	**0.8190**	0.9647	**0.8919**
Spam + Toxic	0.7885	0.9545	0.8715
Twitter + Toxic	0.7973	0.9602	0.8788
Spam + Twitter + Toxic	0.7906	0.9533	0.8720

Table 2: The effect of data sources on Bad KenLM training.

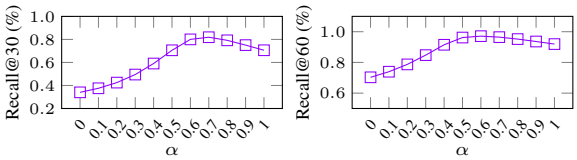


Figure 1: The effect of α on the performance of our ensemble approach.

experiments to determine which of these data sources are effective for training Bad KenLM. In this experiment, we ensembled our Good KenLM with various Bad KenLMs, each trained on different combinations of datasets.

As shown in Table 2, SNS data (Twitter) proved to be the most effective for training Bad KenLM, which is designed to filter out noisy content unsuitable for LLM training. Interestingly, toxic datasets led to a decrease in the performance of Bad KenLM. Unlike SNS data or spam mail, which share similar distributions with web data, toxic datasets contain a large proportion of highly offensive language, resulting in a substantial distributional difference. This discrepancy seems to adversely affect the training process of Bad KenLM.

RQ3: Hyperparameter sensitivity analysis. The parameter α in Eq. (1) adjusts the balance between the PPL scores of Good KenLM and Bad KenLM. We analyze how the performance of our ensemble approach varies with changes in α in terms of Recall@30 and Recall@60.

As depicted in Figure 1, Recall@30 and Recall@60 continuously improve as α increases to 0.7 and 0.6, respectively, and then gradually decrease. These results suggest that when α is too small, the influence of Bad KenLM becomes overly dominant, resulting in poor preservation of high-quality content. Conversely, when α is too large, the influence of Good KenLM prevails, leading to the inclusion of some low-quality content. These results indicate that appropriately determining the value of α is critical for effectively removing noisy content while preserving high-quality content.

RQ4: Degree of computational overhead. To assess the computational overhead of our approach,

Models	Processing Time	Estimated Cost	Throughput	Avg. Recall
Good KenLM	2,234s	$1.42	94.4k docs/s	0.8127
Ens(Good, Bad)	3,928s	$2.50	53.7k docs/s	0.8919

Table 3: Comparison of computational overhead and performance for the CC-MAIN-2024-10 dump between Good KenLM and our ensemble approach.

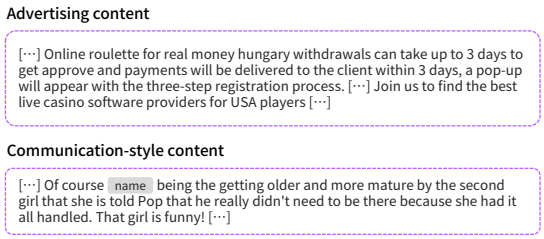


Figure 2: Visualization of examples that are not filtered by Good KenLM but are successfully removed by our ensemble approach.

we measured the processing time and estimated cost[1] for the CC-MAIN-2024-10 dump on a machine with 128-core CPUs. As presented in Table 3, our approach increased the processing time from 2,234 to 3,928 seconds, with an additional cost of $1.08. These increases are justified by the recall improvement from 81.27% to 89.19%, as high-quality data is crucial for effective LLM training.

RQ5: Case study on the effectiveness of our approach. To demonstrate the effectiveness of our ensemble approach, we present examples that are not filtered by Good KenLM but are successfully removed by our ensemble approach. As illustrated in Figure 2, our approach effectively filters advertising and communication-style content, which are generally unsuitable for LLM training. Since advertising content is usually written politely, Good KenLM, trained only on high-quality datasets, struggles to detect it. Conversely, Bad KenLM, trained on spam mail and SNS data, successfully identifies such content as well as communication-style content. Therefore, our ensemble approach more effectively filters these types of content.

5 Conclusion

In this paper, we propose an ensemble approach using Good KenLM and Bad KenLM for effective text filtering. By integrating perplexity scores, we successfully filter out noisy data, such as spam and informal content, while preserving high-quality text. Empirical results suggest that our approach could be a practical solution for filtering large-scale datasets in resource-constrained environments.

[1] It was measured using an AWS r6a.32xlarge (Amazon Web Services, 2022) spot instance.

Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. RS-2024-00338140, Development of learning and utilization technology to reflect sustainability of generative language models and up-to-dateness over time)

Limitations

While the proposed method using Good KenLM and Bad KenLM offers effective filtering of large-scale datasets, it has the following limitations: (i) although our method has demonstrated effectiveness through extensive experiments using Fineweb-edu, we have not been able to measure its direct impact on LLMs training due to computational cost constraints; and (ii) the model relies heavily on pre-defined training datasets, and its performance may degrade when applied to content that significantly differs from the training corpora.

Ethics Statement

The experiments conducted in this paper were carried out objectively and fairly. No biases were introduced during the data selection or evaluation process. All datasets used in this research are publicly available, and the methods were rigorously tested to ensure the reliability and validity of the results.

References

alissonpadua. 2024. Ham spam scam toxic.

Amazon Web Services. 2022. Amazon ec2 r6a instances. Accessed: September 14, 2024.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Together Computer. 2023. Redpajama: an open dataset for training large language models.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

fschatt. 2023. Trump tweets.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Chris J Kennedy, Geoff Bacon, Alexander Sahn, and Claudia von Vacano. 2020. Constructing interval variables via faceted rasch measurement and multitask deep learning: a hate speech application. *arXiv preprint arXiv:2009.10277*.

Xiang Kong, Tom Gunter, and Ruoming Pang. 2024. Large language model-guided document selection. *arXiv preprint arXiv:2406.04638*.

Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. 2024. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36.

lcama. 2022. Elon tweets.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. 2024. Datacomp-lm: In search of the next generation of training sets for language models. *arXiv preprint arXiv:2406.11794*.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection.

Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. 2006. Spam filtering with naive bayes - which naive bayes?

mjw. 2022. sotck market tweets.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A Rossi, and Thien Huu Nguyen. 2023. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *arXiv preprint arXiv:2309.09400*.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2023. Openwebmath: An open dataset of high-quality mathematical web text. *arXiv preprint arXiv:2310.06786*.

John Pavlopoulos, Léo Laugier, Alexandros Xenos, Jeffrey Sorensen, and Ion Androutsopoulos. 2022. From the detection of toxic spans in online discussions to the analysis of toxic-to-civil transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022).*, Dublin, Ireland. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data only. In *Advances in Neural Information Processing Systems*, volume 36, pages 79155–79172. Curran Associates, Inc.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

SciPhi. 2023. Textbooks are all you need : A sciphi collection.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

StephanAkkerman. 2023. Crypto stock tweets.

thehamkercat. 2024. Telegram spam ham.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021. Learning from the worst: Dynamically generated datasets to improve online hate detection. *Preprint*, arXiv:2012.15761.

Xiangpeng Wei, Haoran Wei, Huan Lin, Tianhao Li, Pei Zhang, Xingzhang Ren, Mei Li, Yu Wan, Zhiwei Cao, Binbin Xie, et al. 2023. Polylm: An open source polyglot large language model. *arXiv preprint arXiv:2307.06018*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. QuRating: Selecting high-quality data for training language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 52915–52971. PMLR.

Automatic detection of dyslexia based on eye movements during reading in Russian

Anna Laurinavichyute
University of Potsdam
Department of Linguistics
Karl-Liebknecht-Straße 24-25
14476 Potsdam
laurinavichy@uni-potsdam.de

Anastasiya Lopukhina
Royal Holloway University of London
Department of Psychology
Egham TW20 0EX;
Center for Language and Brain, HSE University
Moscow, Krivokolenny pereulok, 3
Anastasiya.Lopukhina@rhul.ac.uk

David R. Reich
University of Potsdam
Department of Computer Science
An der Bahn 2, 14476 Potsdam;
University of Zurich
Dept of Computational Lingustics
reich@cl.uzh.ch

Abstract

Dyslexia, a common learning disability, requires an early diagnosis. However, current screening tests are very time- and resource-consuming. We present an LSTM that aims to automatically classify dyslexia based on eye movements recorded during natural reading combined with basic demographic information and linguistic features. The proposed model reaches an AUC of 0.93 and outperforms the state-of-the-art model by 7 %. We report several ablation studies demonstrating that the fixation features matter the most for classification.

1 Introduction

One of the most common learning disabilities is dyslexia, a difficulty that specifically affects reading and spelling in individuals with otherwise intact cognitive abilities. The prevalence of dyslexia is estimated to be between 9% and 12% (Katusic et al., 2001; Shaywitz et al., 1998). Early diagnosis is the key factor for getting the needed support and staying on track in the educational system (Glazzard, 2010; Torgesen, 2000; Vellutino et al., 2004).

There are various testing batteries for dyslexia, but most must be administered one-on-one by a trained specialist, who is not always present at school. Moreover, such batteries are still often evaluated using paper-and-pencil methods, which are time-consuming and error-prone. Without a cheap, fast, and reliable mass testing method, the only way to get proper support for a struggling reader is through the educator, who may notice reading difficulties and recommend additional testing.

For mass screening, data should be quick and affordable to obtain. Automatic classification based on (f)MRI and ERP recordings, while suitable from the machine learning perspective, does not fit this criterion. Eye tracking technology is more promising: It provides a rich signal, is unobtrusive, and is growing cheaper. Research on dyslexia and eye movements has a long history. Attempts to tie dyslexia to some characteristics of eye movements go at least as far back as 1980s (Olson et al., 1983; Pavlidis, 1981; Rayner, 1985). Since then, it has been established that the underlying deficit in dyslexia lies not in oculomotor control but rather in phonological decoding (International Dyslexia Association, 2024). Given that eye movements reflect phonological decoding in beginner and adult readers (Rayner et al., 1995, 1998; Blythe, 2014; Leinenger, 2019; Milledge and Blythe, 2019), inferring dyslexia from eye movements is strongly theoretically motivated.

Several machine-learning solutions based on eye movements have been proposed for the mass screening for dyslexia (Asvestopoulou et al., 2019; Nilsson Benfatto et al., 2016; Haller et al., 2022; Jothi Prabha and Bhargavi, 2022; Raatikainen et al., 2021; Rello and Ballesteros, 2015; Shalileh et al., 2023). Yet almost all of these models were trained on very modest samples of 61 (Asvestopoulou et al., 2019) to 185 participants (Nilsson Benfatto et al., 2016). This paper presents a comparison of two models that aim to automatically classify dyslexia on a large dataset comprising eye-movements while reading from 293 young readers of different ages.

1.1 Problem Setting

Inferring whether a child has dyslexia is a binary classification task, and the model's performance can be characterized by a false positive and a true positive rate. By altering the decision threshold, one can observe a receiver operator characteristic curve (ROC curve). The area under the ROC curve (AUC) provides an aggregated measure of performance for all possible classification thresholds.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 59–66
July 27 - August 1, 2025

2 Experiments

2.1 Eye-movement data

The cross-sectional dataset contains eye movements while reading in 293 native speakers of Russian, from the 1st to the 6th grade (published by Shalileh et al. 2023). In Russia, grades 1 through 4 correspond to primary school, and grades 5 and 6 – to secondary school. Based either on a reading assessment or a speech therapist assessment, children were classified as typically developing ($N = 221$) or having developmental dyslexia ($N = 72$). Among children with dyslexia, 43 received their diagnosis based on reading assessment, and 29–based on therapist assessment. Reading assessment was based on the Standardized Assessment of Reading Skills test (SARS, Kornev and Ishimova 2010) and recent normative cut-offs reported by **?**. SARS requires a test-taker to read a short text aloud as quickly and as accurately as possible. The number of words read accurately in the first minute is taken as a measure of reading fluency. If a child scores at least 1.5 standard deviations below their corresponding age mean, a dyslexia label is assigned. Speech therapist assessment is based on a phonological test battery. Note that there are children with dyslexia (diagnosis based on a phonological assessment) whose reading speed is higher than that of some age-matched children in the control group.

All children had nonverbal intelligence scores within the normal range. The typically-developing children had age-appropriate reading fluency and comprehension. Their parents or primary caretakers reported no history of reading disorders. The detailed composition of both groups can be found in Table 1.

2.2 Reading materials

All children were asked to read the same set of 30 sentences comprising the Child Russian Sentence Corpus (Lopukhina et al. 2022). Sentence difficulty was at the level of 3rd to 4th grade, according to an automatic text difficulty measurement developed for Russian (Laposhina and Lebedeva, 2021), and estimated to be 7.42 on the Flesch-Kincaid scale adapted to Russian (Readability Test). Sentences were six to nine words long ($M = 7.6$, $SD = 0.85$). In total, children read 227 words, which contained 182 unique word forms (as words could be repeated across sentences). Individual words were on average 5.6 letters long (range 1–13), and had average lemma frequency of 50.29 items per million (median: 0.73, range: $0.0001 - 667$). The frequency was calculated from the sub-corpus of texts for children of the years 1920–2015 of the Russian National Corpus.

Since Russian is a morphologically rich language, and morphological composition of a word affects its reading time, the number of morphemes comprising each word was annotated. The annotation was first done by an automated parser and then reviewed by a trained linguist. Finally, each word's predictability was estimated using an online cumulative cloze task with 46 children who did not participate in the eye-tracking study. Predictability was measured as the number of correct guesses divided by the total number of guesses. Zero probabilities were replaced with $\frac{1}{2\times46}$, where 46 is the number of guesses for the word.

For a more detailed description of the dataset and reading materials, see Lopukhina et al. 2022.

2.3 Reference method

As a baseline, we use a state-of-the-art (SOTA) SVM-RFE with a linear kernel described and implemented by Haller et al. (2022). This approach was first proposed by Nilsson Benfatto et al. (2016), who reported 96% accuracy on a balanced dataset. As input, the SOTA model uses the means and standard deviations of 12 eye-movement features, such as first fixation duration, first-pass reading time, etc. (for the full list, refer to Haller et al. 2022).

Note that Haller et al. had a homogenous data set of age-matched readers, and did not include age as a predictor. Given that the present dataset includes readers of different ages, we report the performance of the SOTA model both without grade, for full comparability with Haller et al.'s results, and with grade, for a fairer comparison.

2.4 Proposed model

The model introduced by Haller et al. relied on aggregated reading measures. This aggregation, however, results in the loss of significant temporal and word-level information. We address this limitation by employing a sequential method—namely, LSTMs (Hochreiter and Schmidhuber, 1997)—to preserve and leverage temporal information. As demonstrated by previous studies (Ahn et al., 2020; Reich et al., 2022), this choice effectively captures the temporal dynamics necessary for eye-movement classification.

Table 1: Demographic and cognitive characteristics of both participant groups, organized by grade. Values before the slash ("/") represent the control group, while those after the slash correspond to participants with dyslexia.

Grade	**1** (N=50/8)	**2** (N=40/10)	**3** (N=37/20)	**4** (N=39/28)	**5** (N=31/6)	**6** (N=24/0)
Gender Female: N (%)	22 (44%) / 2 (25%)	24 (60%) / 2 (20%)	19 (51%) / 12 (60%)	18 (46%) / 9 (32%)	12 (39%) / 2 (33%)	10 (42%) / -
Age Mean ± SD	$7.32_{0.51}$ / $7.25_{0.46}$	$8.35_{0.48}$ / $8.40_{0.84}$	$9.30_{0.46}$ / $9.30_{0.57}$	$10.18_{0.56}$ / $10.25_{0.59}$	$11.29_{0.78}$ / $11.17_{0.41}$	$12.00_{0.59}$ / -
Nonverbal intelligence Mean ± SD	$29.88_{3.99}$ / $29.75_{4.74}$	$31.00_{3.23}$ / $29.00_{3.74}$	$31.24_{3.50}$ / $31.40_{5.75}$	$31.90_{3.59}$ / $32.14_{3.33}$	$32.81_{2.12}$ / $28.50_{4.85}$	$33.17_{2.39}$ / -
Reading speed (wpm) Mean ± SD	$63.80_{27.06}$ / $17.38_{8.52}$	$79.0_{17.54}$ / $30.70_{10.68}$	$95.57_{13.93}$ / $52.20_{20.48}$	$119.28_{20.67}$ / $57.50_{22.29}$	$122.48_{29.38}$ / $56.50_{16.60}$	$124.62_{23.50}$ / -

Our proposed model's input is a participant's fixation sequence on a sentence. Each input vector consists of basic demographic information, gaze-specific, and linguistic features:

(i) *Demographic features*: participant's age, grade, and gender. Age and grade are relevant for classification because reading skills in primary school are noticeably different between grades, and many reading evaluations are normed for a certain age (grade). Participant's gender is important from a clinical perspective. Boys are diagnosed with dyslexia more often than girls: the male-to-female sex ratio ranges from about 3:1 to 5:1 in self-identified samples, and from 1.5:1 to 3.3:1 in random samples. Arnett et al. (2017) claim that the difference in dyslexia rates between sexes is valid and is driven by lower and more variable processing speed in boys.
(ii) *Gaze-specific features*: fixation duration, fixation horizontal and vertical coordinates on the screen, landing position on the word, next fixation distance, next saccade amplitude, next saccade angle, next saccade velocity, and next saccade direction. We used all fixation features available through the eye movement recording device and reasoned that the importance of different features can be estimated through ablation studies.
(iii) *Linguistic features*: fixated word's length in letters, predictability and frequency (these features explain most of the eye-movement variance, see Kliegl et al. 2004; Shain 2019), as well as number of morphemes comprising the word.

We choose an BiLSTM-based architecture, where the mean of the hidden states is fed into two sequential linear layers, projecting it down to a single sigmoid output to represent the label prediction. Optimized hyperparameters and search space are reported in Appendix A.

3 Results

3.1 Model evaluation

The models are evaluated in two settings: prediction of the reader's status based on a single sentence data (sentence prediction setting) or based on all available reading data (reader prediction setting). In the reader prediction setting, predictions for individual sentences are aggregated to produce the final outcome. The motivation for introducing the sentence prediction setting was to identify whether some sentences serve as a better diagnostic material than others, and to evaluate model precision depending on the amount of input.

All models were evaluated and tuned using 10-fold nested cross-validation and random grid search (see Appendix A). Data from the same person is always constrained to one fold, so that the models always make predictions for unseen participants. The ratio of persons with/without dyslexia is balanced across all folds.[1]

3.2 Results

For all methods, we report AUC for reader- and sentence-level settings (see Table 2). A visual summary of ROC AUC performance can also be found in Figure 1. Classification performance in the reader-prediction setting was numerically higher than in the sentence-prediction setting. However, according to an unpaired one-tailed t-test, the difference between settings was not significant in any model (LSTM: $t(15.55) = 1.22$, $p = 0.12$; $\text{SOTA}_{\text{+Grade}}$: $t(16.21) = 0.81$, $p = 0.21$; $\text{SOTA}_{\text{-Grade}}$: $t(17.83) = 0.24$, $p = 0.41$). The SOTA model that included information on grade performed numerically better, but the difference was not significant (reader prediction setting: $t(17.96) = 1.03$, $p = 0.16$; sentence prediction setting: $t(16.64) = 0.71$, $p = 0.24$). Importantly, the proposed LSTM significantly outperformed

[1]All code is available online: https://github.com/annlaurin/Rus_dyslex_classification

the SOTA$_{+\text{Grade}}$ model in both reader-prediction ($t(12.146) = 2.12$, $p = 0.028$) and sentence-prediction settings ($t(17.92) = 2.20$, $p = 0.021$).

3.2.1 Ablation Studies

In the reader-prediction setting, we run additional ablation studies, assessing model performance without saccade-related measures (next fixation distance, next saccade amplitude, angle, velocity, and direction), without linguistic information (word length, frequency, predictability, and the number of morphemes), without demographic information (age, grade, and gender), and without all eye-movement features. In all ablation studies, AUC score was lower, but the decrease was not significant except for the model without all eye-movement features (LSTM$_{\text{-Saccade}}$: $t(14.70) = 0.95$, $p = 0.17$; LSTM$_{\text{-Ling}}$: $t(17.80) = 0.06$, $p = 0.47$; LSTM$_{\text{-Demographic}}$: $t(16.14) = 1.66$, $p = 0.058$; LSTM$_{\text{-All eye movement}}$: $t(17.93) = 3.25$, $p = 0.002$).

On average, children with dyslexia make more fixations than normally developing children. To evaluate whether the LSTM predictions are based mainly on the input length, we reduced the number of fixations to 29. The resulting AUC score did not differ from the score obtained on the whole input ($t(18) = 0.20$, $p = 0.42$).

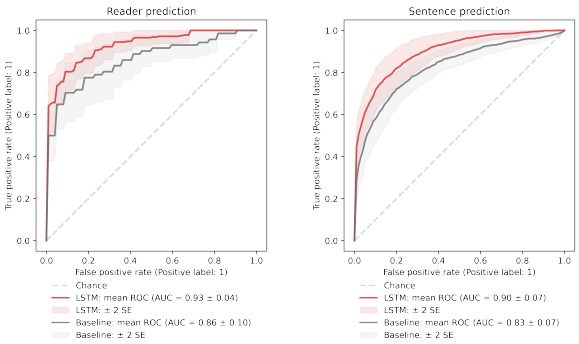


Figure 1: Summary of model performance. SOTA baseline model used grade information.

4 Discussion

The finding that the information on the grade of the reader did not significantly improve the performance of either model is rather surprising. In the present dataset, at least in some cases, dyslexia was diagnosed based on age-specific normative reading speed cut-offs (see Section 2.1). Consequently, information about reader's grade should be crucial for the classification performance. Grade-invariant performance might reflect that the model has captured some invariant property of the eye movements of readers with dyslexia. For the SOTA model trained exclusively on aggregated features, we consider this explanation unlikely. For the LSTM trained on a sequence of fixations, this explanation is more likely, but it is precisely the LSTM that shows a greater numerical decrease in performance without the grade information. In general, we believe that a successful model should be able to uncover the relationship between reading speed, grade, and dyslexia label.

		AUC
Reader	SOTA	$0.86_{\pm 0.10}$
	SOTA$_{\text{-Grade}}$	$0.81_{\pm 0.11}$
	LSTM	$0.93_{\pm 0.05}$
	LSTM$_{\text{-Ling}}$	$0.92_{\pm 0.05}$
	LSTM$_{\text{-Saccade}}$	$0.91_{\pm 0.07}$
	LSTM$_{\text{-Demographic}}$	$0.90_{\pm 0.06}$
	LSTM$_{\text{29 fixations}}$	$0.93_{\pm 0.04}$
	LSTM$_{\text{-All eye movement}}$	$0.87_{\pm 0.04}$
Sentence	SOTA	$0.83_{\pm 0.07}$
	SOTA$_{\text{-Grade}}$	$0.80_{\pm 0.10}$
	LSTM	$0.90_{\pm 0.07}$

Table 2: Summary of AUC $\pm$ standard error in the reader- and sentence-prediction settings.

Another surprising outcome is the lack of difference between the sentence- and reader-prediction settings. Given that the reader-prediction setting relies on $10\times$ to $30\times$ more data, we expected performance to be higher. The difference in performance may not be significant due to the relatively small size of the dataset and insufficient statistical power.

The finding that removing linguistic features did not significantly affect LSTM model's performance is less surprising. Arguably, the most crucial feature for dyslexia classification, word's orthographic transparency (Borleffs et al., 2017), was not available. Including a measure of word orthographic transparency might be a promising next step in improving model performance.

Most importantly, the proposed LSTM outperformed the SOTA model. We can confidently state that better performance is not a trivial result of hyperparameter tuning, as SOTA model was tuned for the same parameters within a similar search space. In the same vein, LSTM did not simply rely on the number of fixations made by children with

and without dyslexia for classification. Based on the outcomes of ablation experiments, we conclude that model performance increased due to the more detailed information about how the sequence of eye movements unfolds.

5 Conclusions

The model of automatic dyslexia detection proposed here has outperformed the SOTA model. Importantly, unlike most of the models proposed so far (Nilsson Benfatto et al., 2016; Haller et al., 2022; Asvestopoulou et al., 2019; Jothi Prabha and Bhargavi, 2022), the present LSTM was trained on an unbalanced dataset of eye movements of children of different ages, and might therefore be more robust and potentially more appropriate for the real-world applications.

Limitations

This decision to include information from participants who did not read all 30 sentences could potentially lead to data leakage: The model may learn that incomplete sessions are more likely to come from a child with dyslexia. We think that this is unlikely for two reasons: First, the proportions of incomplete sessions are not drastically different between the two groups. Second, this potential data leakage should only affect the reader-prediction setting (where the model expects to see 30 sentences), not the sentence-prediction setting (where the model expects to see one sentence). In the present case, there was no significant difference in performance between the reader prediction and the sentence prediction settings (see Section 3.2), so the reader-prediction setting is unlikely to have an unfair advantage.

Ethical considerations

Using demographic variables, such as age and gender, might lead to reproducing existing biases. For example, males are diagnosed with dyslexia more frequently, but at least part of the difference may be attributed to referral bias (Wadsworth et al., 1992). One way to control for bias is to withhold the potentially biasing feature. The ablation experiment that removed the demographic information performed on par with the full model. Therefore, we conclude that the model at least does not enhance the bias that might be present in the data set.

Funding statement

Anna Laurinavichyute was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project ID 317633480 – SFB 1287. The contribution of Anastasiya Lopukhina was supported by the Basic Research Program at the National Research University Higher School of Economics (HSE University). The contribution of David R. Reich was partially funded by the Swiss National Science Foundation under grant IZCOZ0_220330 (EyeNLG).

References

Seoyoung Ahn, Conor Kelton, Aruna Balasubramanian, and Greg Zelinsky. 2020. Towards predicting reading comprehension from gaze behavior. In *ACM Symposium on Eye Tracking Research and Applications*, pages 1–5.

Anne B Arnett, Bruce F Pennington, Robin L Peterson, Erik G Willcutt, John C DeFries, and Richard K Olson. 2017. Explaining the sex difference in dyslexia. *Journal of Child Psychology and Psychiatry*, 58(6):719–727.

Thomais Asvestopoulou, Victoria Manousaki, Antonis Psistakis, Ioannis Smyrnakis, Vassilios Andreadakis, Ioannis M Aslanides, and Maria Papadopouli. 2019. Dyslexml: Screening tool for dyslexia using machine learning. *arXiv preprint arXiv:1903.06274*.

Hazel I Blythe. 2014. Developmental changes in eye movements and visual information encoding associated with learning to read. *Current directions in psychological science*, 23(3):201–207.

Elisabeth Borleffs, Ben AM Maassen, Heikki Lyytinen, and Frans Zwarts. 2017. Measuring orthographic transparency and morphological-syllabic complexity in alphabetic orthographies: a narrative review. *Reading and writing*, 30:1617–1638.

Jonathan Glazzard. 2010. The impact of dyslexia on pupils' self-esteem. *Support for learning*, 25(2):63–69.

Patrick Haller, Andreas Säuberli, Sarah Elisabeth Kiener, Jinger Pan, Ming Yan, and Lena Jäger. 2022. Eye-tracking based classification of Mandarin Chinese readers with and without dyslexia using neural sequence models. *arXiv preprint arXiv:2210.09819*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

International Dyslexia Association. 2024. Dyslexia basics. https://dyslexiaida.org/. Accessed: 2024-07-03.

A Jothi Prabha and R Bhargavi. 2022. Prediction of dyslexia from eye movements using machine learning. *IETE Journal of Research*, 68(2):814–823.

Slavica K Katusic, Robert C Colligan, William J Barbaresi, Daniel J Schaid, and Steven J Jacobsen. 2001. Incidence of reading disability in a population-based birth cohort, 1976–1982, rochester, minn. In *Mayo Clinic Proceedings*, volume 76, pages 1081–1092. Elsevier.

Reinhold Kliegl, Ellen Grabner, Martin Rolfs, and Ralf Engbert. 2004. Length, frequency, and predictability effects of words on eye movements in reading. *European journal of cognitive psychology*, 16(1-2):262–284.

AN Kornev and OA Ishimova. 2010. Metodika diagnostiki disleksii u detey [methods of diagnosis of dyslexia in children]. *St. Petersburg: Publishing house of the Polytechnic University*.

Antonina N Laposhina and Maria Yu Lebedeva. 2021. Textometr: An online tool for automated complexity level assessment of texts for russian language learners. *Russian Language Studies*, 19(3):331–345.

Mallorie Leinenger. 2019. Survival analyses reveal how early phonological processing affects eye movements during reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 45(7):1316.

Anastasiya Lopukhina, Nina Zdorova, Vladislava Staroverova, Nina Ladinskaya, Anastasiia Kaprielova, Sofya Goldina, Olga Vedenina, Ksenia Bartseva, and Olga Dragoy. 2022. Benchmark measures of eye movements during reading in russian children.

Sara V Milledge and Hazel I Blythe. 2019. The changing role of phonology in reading development. *Vision*, 3(2):23.

Mattias Nilsson Benfatto, Gustaf Öqvist Seimyr, Jan Ygge, Tony Pansell, Agneta Rydberg, and Christer Jacobson. 2016. Screening for dyslexia using eye tracking during reading. *PloS one*, 11(12):e0165508.

Richard K Olson, Reinhold Kliegl, and Brian J Davidson. 1983. Dyslexic and normal readers' eye movements. *Journal of Experimental Psychology: Human Perception and Performance*, 9(5):816.

George Th Pavlidis. 1981. Do eye movements hold the key to dyslexia? *Neuropsychologia*, 19(1):57–64.

Peter Raatikainen, Jarkko Hautala, Otto Loberg, Tommi Kärkkäinen, Paavo Leppänen, and Paavo Nieminen. 2021. Detection of developmental dyslexia with machine learning using eye movement data. *Array*, 12:100087.

Keith Rayner. 1985. Do faulty eye movements cause dyslexia? *Developmental Neuropsychology*, 1(1):3–15.

Keith Rayner, Alexander Pollatsek, and Katherine S Binder. 1998. Phonological codes and eye movements in reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 24(2):476.

Keith Rayner, Sara C Sereno, Mary F Lesch, and Alexander Pollatsek. 1995. Phonological codes are automatically activated during reading: Evidence from an eye movement priming paradigm. *Psychological Science*, 6(1):26–32.

Readability Test. Readability test for russian texts. Accessed: 2024-07-04.

David Robert Reich, Paul Prasse, Chiara Tschirner, Patrick Haller, Frank Goldhammer, and Lena A Jäger. 2022. Inferring native and non-native human reading comprehension and subjective text difficulty from scanpaths in reading. In *2022 Symposium on Eye Tracking Research and Applications*, pages 1–8.

Luz Rello and Miguel Ballesteros. 2015. Detecting readers with dyslexia using machine learning with eye tracking measures. In *Proceedings of the 12th international web for all conference*, pages 1–8.

Russian National Corpus. Russian national corpus. Accessed: 2024-07-04.

Cory Shain. 2019. A large-scale study of the effects of word frequency and predictability in naturalistic reading. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, Volume 1 (Long and Short papers)*, pages 4086–4094.

Soroosh Shalileh, Dmitry Ignatov, Anastasiya Lopukhina, and Olga Dragoy. 2023. Identifying dyslexia in school pupils from eye movement and demographic data using artificial intelligence. *Plos one*, 18(11):e0292047.

Sally E Shaywitz, Bennett A Shaywitz, Kenneth R Pugh, Robert K Fulbright, R Todd Constable, W Einar Mencl, Donald P Shankweiler, Alvin M Liberman, Pawel Skudlarski, Jack M Fletcher, et al. 1998. Functional disruption in the organization of the brain for reading in dyslexia. *Proceedings of the National Academy of Sciences*, 95(5):2636–2641.

Joseph K Torgesen. 2000. Individual differences in response to early interventions in reading: The lingering problem of treatment resisters. *Learning disabilities research & practice*, 15(1):55–64.

Frank R Vellutino, Jack M Fletcher, Margaret J Snowling, and Donna M Scanlon. 2004. Specific reading disability (dyslexia): What have we learned in the past four decades? *Journal of child psychology and psychiatry*, 45(1):2–40.

Sally J Wadsworth, JC DeFries, Jim Stevenson, Jeffrey W Gilger, and BF Pennington. 1992. Gender ratios among reading-disabled children and their siblings as a function of parental impairment. *Journal of Child Psychology and Psychiatry*, 33(7):1229–1239.

A Model parameters

Model search space is summarized in Table 3.

Batch size	8, 16, 32, 64, 128
Learning rate	$15 \times \mathcal{U} \sim (1.0 \times 10^{-5}, 1.0 \times 10^{-1})$
LSTM hidden layer size	30, 40, 50, 60, 70

Table 3: Hyperparameter search space.

The optimal parameters can be found in Table 4.

Batch size	Learning rate	Hidden layer size
	Reader prediction setting	
64	0.001	40
16	0.001	40
64	0.01	30
64	0.001	40
64	0.001	40
64	0.001	40
16	0.01	30
32	0.01	30
16	0.01	50
64	0.001	40
	Sentence prediction setting	
8	7.07×10^{-5}	30
8	0.0003	50
128	4.21×10^{-5}	70
64	0.0025	50
8	7.07×10^{-5}	30
64	0.0025	50
128	0.0025	30
8	7.07×10^{-5}	30
32	5.34×10^{-5}	70
8	4.21×10^{-5}	50

Table 4: Resulting optimal parameters.

Doc-React: Multi-page Heterogeneous Document Question-answering

Junda Wu[1], Yu Xia[1], Tong Yu[2], Xiang Chen[2], Sai Sree Harsha[2], Akash V. Maharaj[2], Ruiyi Zhang[2], Victor Bursztyn[2], Sungchul Kim[2], Ryan Rossi[2], Julian McAuley[1], Yunyao Li[2], Ritwik Sinha[2]
[1]University of California San Diego [2]Adobe Inc.

Abstract

Answering questions over multi-page, multimodal documents, including text and figures, is a critical challenge for applications that require answers to integrate information across multiple modalities and contextual dependencies. Existing methods, such as single-turn retrieval-augmented generation (RAG), struggle to retrieve fine-grained and contextually relevant information from large, heterogeneous documents, leading to suboptimal performance. Inspired by iterative frameworks like ReAct, which refine retrieval through feedback, we propose Doc-React, an adaptive iterative framework that balances information gain and uncertainty reduction at each step. Doc-React leverages InfoNCE-guided retrieval to approximate mutual information, enabling dynamic sub-query generation and refinement. A large language model (LLM) serves as both a judge and generator, providing structured feedback to iteratively improve retrieval. By combining mutual information optimization with entropy-aware selection, Doc-React systematically captures relevant multimodal content, achieving strong performance on complex QA tasks.

1 Introduction

Answering queries over multi-page, multimodal documents with textual and visual information is a critical challenge (Ma et al., 2024a; Tanaka et al., 2023). These documents, which are typically long, often have diverse layouts and interleaved content (Nguyen et al., 2024), requiring reasoning across modalities. Specifically, as illustrated by the use case in Figure 1, the input is a user query and multi-page documents, either provided by the user or from a database. The goal is to produce an accurate and concise text answer by analyzing the relevant content across pages.

Existing methods (Yu et al., 2024; Yang et al., 2023; Faysse et al., 2024; Yao et al., 2022; Wu et al., 2024d) retrieve multimodal content and feed it into single-turn RAG models. While effective in simpler scenarios, these methods fall short in multi-page settings, missing fine-grained, contextually relevant information. As illustrated in Figure 1, when applying (Yu et al., 2024; Faysse et al., 2024), the query Q does not explicitly mention the region with "252M mobile broadband subscriptions." Simply retrieving figures semantically similar to Q fails to identify that the region refers to North America and further locate the figure containing North America's "active social network users" (described in S_2). This highlights the need for an iterative approach to dynamically refine queries, discover relevant figures, and gather necessary information. Inspired by iterative methods (Yao et al., 2022), we adopt a dynamic approach to refine queries iteratively, addressing the challenges of multi-page document QA.

We propose Doc-React, an adaptive iterative framework for multi-page document retrieval and acting. Doc-React refines retrieval and query formulation in distinct steps to maximize normalized information gain, balancing information gain and uncertainty reduction with InfoNCE-guided retrieval. It addresses residual gaps through sub-queries refined by an LLM, which provides feedback as both judge and generator. By maximizing step-wise information gain, Doc-React iteratively captures fine-grained, contextually relevant information, adapting queries and improving retrieval efficiency. In summary, our contributions are:

- We propose Doc-React, an adaptive iterative framework that dynamically refines multimodal retrieval and QA performance through LLM-guided feedback.
- We introduce a method to balance information gain and uncertainty reduction, dynamically adapting sub-queries and retrieval prompts.
- We conduct extensive experiments showing

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 67–78
July 27 - August 1, 2025

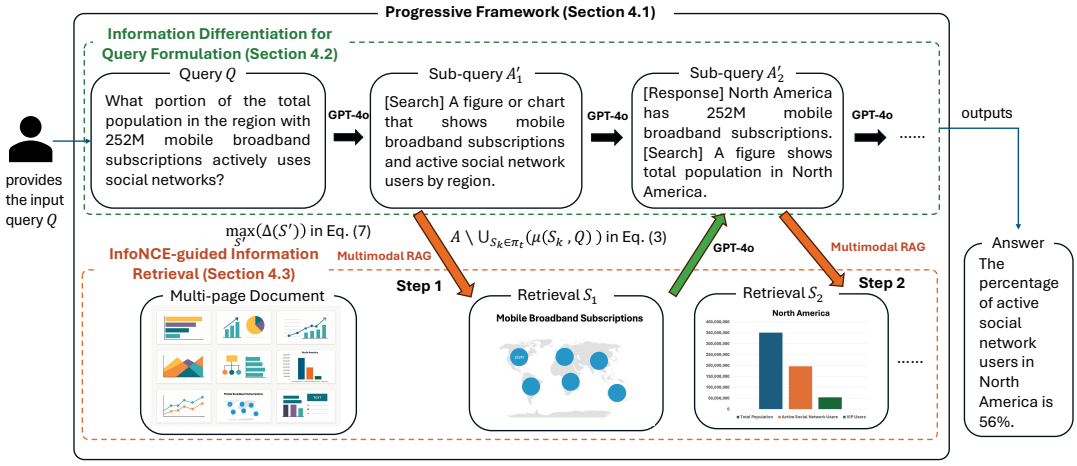


Figure 1: Doc-React applied to the multi-page document QA task. The framework processes a user query as input and operates on multi-page documents. It iteratively refines information retrieval and query formulation to maximize information gain and reduce uncertainty, ultimately generating an accurate and contextually relevant answer.

significant improvements over baselines for complex multi-page QA tasks.

2 Related Work

2.1 Iterative and Adaptive Retrieval

Iterative approaches, such as ReAct (Yao et al., 2022), Chain-of-thought (Wei et al., 2022; Wu et al., 2024c,a) and MM-React (Yang et al., 2023), leverage dynamic feedback mechanisms to refine queries and align context (Shinn et al., 2024; Huang et al., 2023; Zhang et al., 2024), effectively addressing challenges in noisy or incomplete retrieval scenarios (Zhou et al., 2024). These methods demonstrate the importance of iterative frameworks (Hu et al., 2024b; Yang et al., 2024) in achieving more accurate and context-aware retrieval. Similarly, mutual information optimization techniques, inspired by InfoNCE (Oord et al., 2018), rely on iterative strategies to balance information gain and uncertainty reduction, further highlighting the relevance of iterative methods for handling complex retrieval tasks. In our framework, InfoNCE-guided mutual information optimization is not only theoretically motivated but also serves as a practical mechanism for adaptive multi-step retrieval (detailed in Section 4).

2.2 Multimodal Document Retrieval

Advancements in multimodal learning enable systems to jointly process text and visual information (Yan et al., 2024; Wu et al., 2025, 2024b; Yao et al., 2024; Liu et al., 2024a). Models like LayoutLM (Xu et al., 2020b,a) leverage spatial embeddings for document layout understanding, excelling at tasks like form and invoice parsing. These models are limited to single-page documents (Liu et al., 2024b) and cannot handle reasoning over multi-page (Ma et al., 2024a), and heterogeneous layouts (Tanaka et al., 2023).

2.3 Retrieval-Augmented Generation (RAG)

RAG (Lewis et al., 2020) and FiD (Izacard and Grave, 2020) combine retrieval and generation for open-domain QA. While effective, their reliance on static pipelines limits adaptability to complex document layouts. Retrieval methods like BM25 (Robertson and Walker, 1994) and dense models such as ColBERT (Khattab and Zaharia, 2020) efficiently match content but fail to capture the cross-modal dependencies for multimodal documents. MuRAR (Zhu et al., 2024) generates multimodal responses to convey rich cross-modal information.

3 Formulation: Multi-page Doc-QA

We introduce a novel formulation for the multi-page document question-answering (Doc-QA) task, where the input consists of heterogeneous documents that interleave images and texts in unstructured, free-form layouts. This practical task involves disentangling multimodal information embedded within the images of document pages to accurately answer a given query. The objective of a multi-page Doc-QA is to extract sufficient multimodal information from the relevant pages of a document and use it to generate a correct answer. Our novel formulation is different from the formulation of existing works (Mathew et al., 2021; Chang et al., 2022; Masry et al., 2022; Liu et al.,

2024b) that we consider question-answering with a long sequence of document page images due to heterogeneous layouts of various types of documents. Different from tool-chain based agentic framework (Yang et al., 2023; Ma et al., 2024b; Kumar et al., 2024), our formulation focuses on balancing between information gain and generation uncertainty when multiple images are in the prompts.

Given a N-page document $D = \{p_1, p_2, \ldots, p_N\}$ and a query Q, an underlying policy is required to identify multiple groups of document pages $\pi = \{S_i | S_i \subseteq D\}$, where each group of document pages S_i provides relevant information to a specific aspect of the query Q. The multimodal information necessary to derive the answer A must then be extracted from each subset S_i, conditioned on the query Q, using a multimodal large language model (MLLM) denoted as $\mu(S_i, Q)$. The final output is a text-only response, generated based on the extracted multimodal content. To quantify the informativeness of this process, we measure the mutual information $I(\mu(S_i, Q); A)$ as the amount of information required to determine the answer A.

We further formalize this task as an information optimization problem, which seeks a balance between reducing uncertainty and ensuring information completeness (Alemi et al., 2016):

$$\begin{aligned} \min_{I} \quad & \sum_{S_i \in \pi} H(\mu(S_i, Q)), \quad S_i \subseteq D, \quad \forall S_i \in \pi, \\ \text{s.t.} \quad & I(\bigcup_{S_i \in \pi} \mu(S_i, Q); A) \geq I(\mu(D, Q); A), \end{aligned} \tag{1}$$

where $H(\cdot)$ denotes the entropy, capturing uncertainty in the extracted information. A key practical challenge is the increasing estimation error as the number of input images grows. Empirically, we can observe that there exist subsets $S_1, S_2 \subseteq D$ such that $H(\mu(S_1, S_2, Q)) > H(\mu(S_1, Q)) + H(\mu(S_2, Q))$. This implies that increasing the size of the input set does not always improve answer accuracy, while the increased input size also retrains the usage of MLLMs, due to their limited perception of multimodal context.

4 Method: Doc-React

The formulation's major challenge is incrementally identifying and aggregating multimodal information from a multi-page document corpus (Hu et al., 2024a; Cho et al., 2024). Inspired by information-entropy optimization (Alemi et al., 2016; Still, 2009), we propose **Doc-React**, which greedily maximizes the entropy-normalized information gain for each round of retrieval and action.

Algorithm 1 Doc-React: Adaptive Retrieval with Normalized MI Gain Maximization

Require: Document $D = \{p_1, p_2, \ldots, p_N\}$, Query Q, MLLM μ, Retrieval model $f(\cdot, \cdot)$, Differentiation prompt P_{diff}, Maximum iteration T, Candidate pool Ω.

Initialize $\pi_0 \leftarrow \emptyset$.
for $t = 0$ to $T - 1$ **do**
  Sub-query Formulation with Residual Information Approximation:
  Formulate sub-queries according to Eq.(5);
  Candidate Evaluation:
  for each $S' \in \Omega$ **do**
    Estimate the lower bound of the mutual information gain according to Eq.(6);
    Compute entropy $H(\mu(S', Q))$ from MLLM's generation distributions;
    Compute normalized gain $\Delta(S')$ according to Eq.(4);
  end for
  Select Best Subset:
  $S^*_{t+1} \leftarrow \arg\max_{S' \in \Omega'} \Delta(S')$
  Update:
  $\pi_{t+1} \leftarrow \pi_t \cup \{S^*_{t+1}\}$
  if stopping criteria met **then**
    break
  end if
end for
return Final selected subsets π_T, and final retrieval results $\mu(\{S_k\}_{k \in \pi_T}, Q)$.

4.1 Progressive Framework Refining Retrieval and Query Formulation

At time step t, we maintain a selection of document page subsets π_t, where each subset's $S_k \in \pi_t$ multimodal information is extracted by the MLLM $\mu(S_k, Q)$. To solve the information optimization problem in Eq.(1), we propose to maximize the *normalized information gain* for the next step $t+1$:

$$\Delta(S_{t+1}) = \frac{I\left(\mu(S_{t+1}, Q); A \mid \bigcup_{S_k \in \pi_t} \mu(S_k, Q)\right)}{H(\mu(S_{t+1}, Q))}, \tag{2}$$

where A represents the ground-truth answer to the query. However, different from conventional goal-oriented optimization tasks (Sutton, 2018; Levine et al., 2016), which assume a good representation

of the goal, our task poses a challenge of lacking such *oracle information*.

4.2 Information Differentiation

For query formulation, we propose an information differentiation process to approximate each step's maximum information gain,

$$A'_t = A \setminus \bigcup_{S_k \in \pi_t} \mu(S_k, Q), \tag{3}$$

where A'_t denotes the information residual at time step t. Takes Eq.(3) into Eq.(2), we can derive the approximation of the original information gain,

$$\Delta(S_{t+1}) = \frac{I(\mu(S_{t+1}, Q); A'_t)}{H(\mu(S_{t+1}, Q))}, \tag{4}$$

where the residual information is practically approximated by the MLLM

$$\tilde{A}'_t = \mu(\{S_k\}_{k \in \pi_t}, Q, P_{\text{diff}}), \tag{5}$$

which aggregates retrieved information $\{S_k\}_{k \in \pi_t}$ and the query Q, quantifying the information gap.

Theorem 1 (**Information Gain Derivation via Information Differentiation**). *With the information differentiation enabled as A'_t (in Eq.(3)), the information gain $\Delta(S_{t+1})$ in Eq.(2) can be derived via the information differentiation in Eq.(4).*

We prove the equivalency between Eq.(2) and Eq.(4) in Appendix D. Then, this gap is addressed by leveraging the LLM-as-a-judge (Zheng et al., 2023; Rosset et al., 2024; Gu et al., 2024) to evaluate and decompose complex queries (Xia et al., 2024). Specifically, the gap is formulated into sub-queries by following the in-context examples (in Appendix E).

4.3 InfoNCE-guided Information Retrieval

To further track the information gain $I(\mu(S_{t+1}, Q); A'_t)$, we introduce existing multimodal retrieval models (Faysse et al., 2024; Yu et al., 2024) as prior, which can efficiently provide such information quantification through a retrieval process. Given the multimodal retrieval model $f(\cdot, \cdot)$ which measures the distance between multimodal inputs, we can derive the variational lower bound for the mutual information estimation following InfoNCE (Oord et al., 2018),

$$\begin{aligned} I(\mu(S_{t+1}, Q); A'_t) &\geq -\mathcal{L}_{\text{InfoNCE}}(S_{t+1}) \\ &= \mathbb{E}\left[\log \frac{\exp(f(S_{t+1}, [Q, A'_t]))}{\frac{1}{|\Omega|}\sum_{S' \in \Omega} \exp(f(S', [Q, A'_t]))}\right], \end{aligned} \tag{6}$$

Dataset		MMLongBench		SlideVQA	
Method	Model	F1	Acc	F1	Acc
VisRAG	GPT-4o	29.02	27.78	48.37	46.93
	InternVL	11.43	11.03	18.13	17.28
	MiniCPM	8.28	8.17	12.62	11.83
ColPali	GPT-4o	32.17	30.76	53.42	52.33
	InternVL	8.86	8.57	16.34	15.23
	MiniCPM	7.84	7.74	13.34	12.48
DocReact	GPT-4o	**38.07**	**38.29**	**54.87**	**55.04**

Table 1: Comparison with multimodal retrieval-augmented generation baselines.

Dataset		MMLongBench		SlideVQA	
Method	Model	F1	Acc	F1	Acc
Standard (i.e., w/o CoT)	GPT-4o	**42.00**	**41.71**	53.58	53.41
	InternVL	4.38	4.31	19.00	18.20
	MiniCPM	5.59	5.55	13.85	13.03
CoT	GPT-4o	41.17	40.77	52.12	51.71
	InternVL	10.79	10.47	17.45	16.74
	MiniCPM	6.37	6.13	12.45	11.73
DocReact	GPT-4o	38.07	38.29	**54.87**	**55.04**

Table 2: Comparisons with multi-image multimodal LLM baselines.

where $\Omega = \{S' \mid \forall S' \subseteq D\}$, while in practice we select from a random subset Ω' to reduce the computation complexity. In addition, the estimation uncertainty $H(\mu(S_{t+1}, Q))$ can be calculated based on the MLLM's generation probabilities.

Therefore, using Eq.(6) in Eq.(2), we optimize step-wise normalized information gain $\Delta(S_{t+1})$, by maximizing its lower-bound estimation,

$$\begin{aligned} S^*_{t+1} &= \arg\max_{S' \in \Omega} \Delta(S'), \\ \pi_{t+1} &\leftarrow \pi_t \cup \{S^*_{t+1}\}, \end{aligned} \tag{7}$$

for which, we can achieve the near-optimal solution to the original problem in Eq.(1). We illustrate our method in Algorithm 1.

5 Experiments

For our comparison experiments, we use two lines of baselines, including **Multi-image MLLMs:** GPT4-o, InternVL (Chen et al., 2024), and MiniCPM (Yao et al., 2024), suitable for multi-page inputs; and **Multimodal RAG:** ColPali (Faysse et al., 2024) and VisRAG-Ret (Yu et al., 2024). To evaluate baselines and DocReact on multi-page document question-answering, we apply our method on two datasets, **SlideVQA** (Tanaka et al., 2023) and **VisualWebBench** (Liu et al., 2024b). For details, please refer to Appendix C.

Method	MMLongBench		SlideVQA	
	F1	Acc	F1	Acc
ColPali	32.17	30.76	53.42	52.33
DocReact (w/o ColPali)	37.22	37.37	54.12	54.29
DocReact (w/ ColPali)	**38.07**	**38.29**	**54.87**	**55.04**

Table 3: Ablation study comparing Doc-React with and without ColPali retrieval. Results are reported on the MMLongBench and SlideVQA benchmarks.

Comparison with Multimodal RAG We evaluate DocReact's effectiveness against multimodal RAG methods in Table 1. While multimodal RAG methods mitigate input information overload, they struggle with complex queries requiring progressive multi-image reasoning and retrieval. In contrast, DocReact addresses these challenges through step-wise information differentiation and retrieval, achieving strong performance on both datasets.

Comparison with Multi-image MLLMs We evaluate DocReact's efficiency in processing multi-page document inputs compared to multi-image MLLMs. As shown in Table 2, GPT-4o performs well on MMLongBench, where the limited number of candidate images allows it to process the full context effectively. However, *simply relying on GPT-4o (i.e., the baseline Standard) is not scalable for real-world scenarios with a larger number of candidate images*, where DocReact's iterative approach becomes essential for efficiently retrieving fine-grained, contextually relevant information. While CoT reasoning offers limited improvement on GPT-4o due to inefficiencies in handling multi-modal inputs, DocReact achieves competitive performance on MMLongBench and surpasses GPT-4o on the SlideVQA dataset, showing its scalability and effectiveness in handling multi-page QA tasks.

Case Study We validate DocReact's advantage by a case study (Appendix A). In this example, the query Q does not explicitly mention the region with "252M mobile broadband subscriptions." Advanced retrieval methods such as ColPali (Faysse et al., 2024) fail to infer that the region refers to North America and subsequently miss the figure containing relevant information on active social network users. In contrast, DocReact dynamically refines its search process. Through intermediate reasoning steps (Thought 2 and Thought 3), it identifies the missing region, retrieves the relevant figure, and gets the information (Observation 3), ultimately yielding the accurate result of 56% (Action 4).

Ablation Study on the Effect of Retriever Instead of relying solely on the strong retrieval approach ColPali (Faysse et al., 2024; Yao et al., 2024; Chen et al., 2024), we conducted additional experiments to compare **DocReact (w/ ColPali)** and **DocReact (w/o ColPali)**. In the latter setting, we replaced ColPali with a weaker retrieval method as indicated in Table 3. This comparison allows us to better understand whether the observed improvements are attributable to our framework itself, rather than being solely due to the strong retrieval backbone. Our experiments demonstrate that even without leveraging the strong ColPali method, our approach still outperforms directly using ColPali.

6 Conclusion

We address the challenge of answering questions over multi-page, multimodal documents by introducing Doc-React, an adaptive iterative framework for retrieval and acting. By balancing information gain and uncertainty reduction at each step, the framework leverages InfoNCE-guided retrieval and a large language model (LLM) for dynamic query refinement. Through systematic optimization and feedback, Doc-React effectively captures fine-grained and contextually relevant information, and achieves strong performance.

7 Limitations

Our approach maximizes the normalized mutual information gain, which is achieved through an iterative framework that approximates this objective (as discussed in the theoretical analysis in Appendix D). While the framework effectively refines retrieval and query formulation, the approximation process may lead to slight inefficiencies in specific cases. Future work could focus on refining the optimization process to further streamline the approximation.

References

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*.

Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. 2022. Webqa: Multihop and multimodal qa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16495–16504.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. 2024. M3docrag: Multimodal retrieval is what you need for multi-page multi-document understanding. *arXiv preprint arXiv:2411.04952*.

Manuel Faysse, Hugues Sibille, Tony Wu, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2024. Colpali: Efficient document retrieval with vision language models. *arXiv preprint arXiv:2407.01449*.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, et al. 2024a. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. *arXiv preprint arXiv:2403.12895*.

Xiang Hu, Hongyu Fu, Jinge Wang, Yifeng Wang, Zhikun Li, Renjun Xu, Yu Lu, Yaochu Jin, Lili Pan, and Zhenzhong Lan. 2024b. Nova: An iterative planning and search approach to enhance novelty and diversity of llm generated ideas. *arXiv preprint arXiv:2410.14255*.

Di Huang, Ziyuan Nan, Xing Hu, Pengwei Jin, Shaohui Peng, Yuanbo Wen, Rui Zhang, Zidong Du, Qi Guo, Yewen Pu, et al. 2023. Anpl: towards natural programming with interactive decomposition. *Advances in Neural Information Processing Systems*, 36:69404–69440.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Somnath Kumar, Yash Gadhia, Tanuja Ganu, and Akshay Nambi. 2024. Mmctagent: Multi-modal critical thinking agent framework for complex visual reasoning. *arXiv preprint arXiv:2405.18358*.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. 2016. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024a. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Junpeng Liu, Yifan Song, Bill Yuchen Lin, Wai Lam, Graham Neubig, Yuanzhi Li, and Xiang Yue. 2024b. Visualwebbench: How far have multimodal llms evolved in web page understanding and grounding? *arXiv preprint arXiv:2404.05955*.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. 2024a. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *arXiv preprint arXiv:2407.01523*.

Zixian Ma, Jianguo Zhang, Zhiwei Liu, Jieyu Zhang, Juntao Tan, Manli Shu, Juan Carlos Niebles, Shelby Heinecke, Huan Wang, Caiming Xiong, et al. 2024b. Taco: Learning multi-modal action models with synthetic chains-of-thought-and-action. *arXiv preprint arXiv:2412.05479*.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Dang Nguyen, Jian Chen, Yu Wang, Gang Wu, Namyong Park, Zhengmian Hu, Hanjia Lyu, Junda Wu, Ryan Aponte, Yu Xia, et al. 2024. Gui agents: A survey. *arXiv preprint arXiv:2412.13501*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, pages 232–241. Springer.

Corby Rosset, Ho-Lam Chung, Guanghui Qin, Ethan C Chau, Zhuo Feng, Ahmed Awadallah, Jennifer Neville, and Nikhil Rao. 2024. Researchy questions: A dataset of multi-perspective, decompositional questions for llm web agents. *arXiv preprint arXiv:2402.17896*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Susanne Still. 2009. Information-theoretic approach to interactive learning. *Europhysics Letters*, 85(2):28005.

Richard S Sutton. 2018. Reinforcement learning: An introduction. *A Bradford Book*.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. 2023. Slidevqa: A dataset for document visual question answering on multiple images. In *AAAI*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Junda Wu, Xintong Li, Ruoyu Wang, Yu Xia, Yuxin Xiong, Jianing Wang, Tong Yu, Xiang Chen, Branislav Kveton, Lina Yao, et al. 2024a. Ocean: Offline chain-of-thought evaluation and alignment in large language models. *arXiv preprint arXiv:2410.23703*.

Junda Wu, Xintong Li, Tong Yu, Yu Wang, Xiang Chen, Jiuxiang Gu, Lina Yao, Jingbo Shang, and Julian J McAuley. 2024b. Commit: Coordinated instruction tuning for multimodal large language models. *CoRR*.

Junda Wu, Yuxin Xiong, Xintong Li, Yu Xia, Ruoyu Wang, Yu Wang, Tong Yu, Sungchul Kim, Ryan A Rossi, Lina Yao, et al. 2025. Mitigating visual knowledge forgetting in mllm instruction-tuning via modality-decoupled gradient descent. *arXiv preprint arXiv:2502.11740*.

Junda Wu, Tong Yu, Xiang Chen, Haoliang Wang, Ryan Rossi, Sungchul Kim, Anup Rao, and Julian McAuley. 2024c. Decot: Debiasing chain-of-thought for knowledge-intensive tasks in large language models via causal intervention. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14073–14087.

Junda Wu, Zhehao Zhang, Yu Xia, Xintong Li, Zhaoyang Xia, Aaron Chang, Tong Yu, Sungchul Kim, Ryan A Rossi, Ruiyi Zhang, et al. 2024d. Visual prompting in multimodal large language models: A survey. *arXiv preprint arXiv:2409.15310*.

Yu Xia, Junda Wu, Sungchul Kim, Tong Yu, Ryan A Rossi, Haoliang Wang, and Julian McAuley. 2024. Knowledge-aware query expansion with large language models for textual and relational retrieval. *arXiv preprint arXiv:2410.13765*.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. 2020a. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020b. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1192–1200.

An Yan, Zhengyuan Yang, Junda Wu, Wanrong Zhu, Jianwei Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Julian McAuley, Jianfeng Gao, et al. 2024. List items one by one: A new data source and learning paradigm for multimodal llms. *arXiv preprint arXiv:2404.16375*.

Yujing Yang, Boqi Chen, Kua Chen, Gunter Mussbacher, and Dániel Varró. 2024. Multi-step iterative automated domain modeling with large language models. In *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*, pages 587–595.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, et al. 2024. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. *arXiv preprint arXiv:2410.10594*.

Xianren Zhang, Xianfeng Tang, Hui Liu, Zongyu Wu, Qi He, Dongwon Lee, and Suhang Wang. 2024. Divide-verify-refine: Aligning llm responses with complex instructions. *arXiv preprint arXiv:2410.12207*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot

arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Zhanke Zhou, Rong Tao, Jianing Zhu, Yiwen Luo, Zengmao Wang, and Bo Han. 2024. Can language models perform robust reasoning in chain-of-thought prompting with noisy rationales? *arXiv preprint arXiv:2410.23856*.

Zhengyuan Zhu, Daniel Lee, Hong Zhang, Sai Sree Harsha, Loic Feujio, Akash Maharaj, and Yunyao Li. 2024. Murar: A simple and effective multimodal retrieval and answer refinement framework for multimodal question answering. *arXiv preprint arXiv:2408.08521*.

A Case Study

In this section, we illustrate a case study about the comparison between DocReact and CoT baseline. The desired answer, according to the ground truth, is 56%. We show in the use case that DocReact can successfully enable information verbal differentiation by identifying the missing information in the current observation (highlighted in green font). In addition, DocReact can further formulate sub-queries based on the information differentiation results, which can be used to retrieve more document image potentially containing such information (highlighted in orange font).

Question: What percentage of the TOTAL POPULATION in the region where there are 252M mobile broadband subscriptions are ACTIVE SOCIAL NETWORK USERS?

(i) The result by our DocReact

Thought 1: I need to find the region where there are 252M mobile broadband subscriptions and then determine the percentage of the total population in that region who are active social network users.

Action 1: Search[A figure or chart that shows mobile broadband subscriptions and active social network users by region.]

Observation 1: 6,572,950,124 mobile subscriptions

Thought 2: The observation does not provide the necessary information about the region with 252M mobile broadband subscriptions. I need to search again for a figure or chart that shows mobile broadband subscriptions and active social network users by region.

Action 2: Search[A figure or chart that shows mobile broadband subscriptions and active social network users by region, specifically looking for 252M mobile broadband subscriptions.]

Observation 2: North America has 252M mobile broadband subscriptions

Thought 3: I need to search for a figure or chart that shows active social network users in North America.

Action 3: Search[A figure or chart that shows active social network users in North America.]

Observation 3: 56% North America's share of active social network

Action 4: Finish[56%]

(ii) The result by the baseline Standard

The image does not provide a direct percentage of active social network users in the region with 252M mobile broadband subscriptions. However, it does show that there are 173,871,400 active social network users in the region with 252M mobile broadband subscriptions. To calculate the percentage, we would need the total population of that region, which is not provided in the image.

(iii) The result by the baseline CoT

The image shows a bar graph with different regions and their respective mobile broadband subscriptions and active social network users. To find the percentage of active social network users, we need to look at the bar corresponding to the region with 252M mobile broadband subscriptions. The bar indicates that the percentage of active social network users in that region is 17%. Therefore, the answer is 17%.

B Implementation Discussions

While our approach may involve more API calls than baseline methods due to its iterative nature, it is important to note that iterative frameworks are both necessary and common for addressing complex retrieval tasks. Prior works, such as ReAct (Yao et al., 2022), have demonstrated the effectiveness of dynamic feedback mechanisms for refining queries in noisy or incomplete scenarios. Similarly, mutual information optimization methods inspired by InfoNCE (Oord et al., 2018) rely on iterative strategies to balance information gain and uncertainty reduction effectively.

Doc-React builds on these principles by leveraging an iterative query refinement framework to progressively retrieve the most relevant information. This approach significantly reduces the number of input tokens processed per API call by avoiding redundant or irrelevant content. Consequently, while iterative, Doc-React achieves a lower overall computational cost in terms of processed tokens compared to baseline methods, which often rely on processing larger, less refined multimodal contexts in a single pass. The iterative design is not only practical but also essential for tackling the complexity of multi-page, multimodal document QA tasks. Given the markedly inferior performance of open-sourced MLLMs (e.g., MiniCPM and InternVL) on

such challenging tasks, we opt to build DocReact on the stronger baseline GPT-4o to enable a more rigorous comparison. In practice, we limit the process to a maximum of 4 iterations of retrieval and action steps. The MLLM has the flexibility to terminate early by directly outputting the final answer. However, if the MLLM does not voluntarily end the interaction, the system enforces termination after the final iteration. At this point, the answer is generated using aggregated information from all previous rounds.

C Experiment Details

C.1 Dataset Details

We evaluate our method on the following datasets:

- **SlideVQA** (Tanaka et al., 2023): Designed for structured documents like presentation slides, focusing on reasoning with visual elements such as diagrams and charts.
- **MMLongBench-Doc** (Ma et al., 2024a): Evaluates tasks requiring long-context reasoning over multimodal documents.

C.2 Baseline Details

We include the following baselines in our comparisons:

- **Multi-image MLLMs:** GPT4-o, InternVL (Chen et al., 2024), and MiniCPM (Yao et al., 2024).
- **Multimodal RAG:** ColPali (Faysse et al., 2024) and VisRAG-Ret (Yu et al., 2024).

C.3 Implementation Details

- **Framework:** Implemented using the Hugging Face Transformers library for integration with retrieval and generative models.
- **Query Refinement:** Iterative query refinement is driven by LLM feedback and InfoNCE-based optimization.
- **Hyperparameters:** We set generation temperature as 0 for deterministic decoding. We set the token limit of 100 for generation and the maximum number of images input as 50 for multi-image MLLM baselines.
- **Evaluation Metrics:** We follow the evaluation metrics introduced by MMLongBench-Doc (Ma et al., 2024a) where Acc and F1 are used to meature QA performance.
- **Hardware:** Experiments are conducted on 4 NVIDIA A6000 GPUs with multi-GPU support.

D Theoretical Analysis

This section provides more details and the proof of Theorem 1.

Proof. Denote the accumulated information as

$$Z_t = \bigcup_{S_k \in \pi_t} \mu(S_k, Q).$$

We can assume the information differentiation by

$$A = g(Z) + A'_t, \tag{8}$$

where $g(Z)$ is the part of A explained by Z. By definition of conditional mutual information, the information gain term in Eq.(2) can be derived

$$\begin{aligned} I(\mu(S_{t+1}, Q); A \mid Z) = H(A \mid Z) \\ - H(A \mid \mu(S_{t+1}, Q), Z). \end{aligned}$$

Using the decomposition assumption (in Eq.(8)) of information differentiation, we show that

$$\begin{aligned} H(A \mid Z) &= H(A'_t \mid Z), \\ H(A \mid \mu(S_{t+1}, Z) &= H(A'_t \mid \mu(S_{t+1}, Z). \end{aligned}$$

In addition, since the differentiated information A'_t is assumed to be independent from the currently collected information Z_t, we can further reduce the results as

$$\begin{aligned} &H(A'_t \mid Z) = H(A'_t), \\ &H(A'_t \mid \mu(S_{t+1}, Z) = H(A'_t \mid \mu(S_{t+1}). \end{aligned}$$

Therefore, we could conclude with the equivalency

$$\begin{aligned} I(\mu(S_{t+1}; A \mid Z) &= H(A'_t) - H(A'_t \mid \mu(S_{t+1})) \\ &= I(\mu(S_{t+1}); A'_t). \end{aligned}$$

□

E In-context Examples

We illustrate the three in-context learning examples we used to align MLLMs' behaviors in generating thoughts of information differentiation (**Thought** t), actions of multimodal search sub-queries (**Action** t), and MLLM received feedback back through MLLM information retrieval (**Observation** t).

Example 1:

Question: What is the elevation range for the area that the eastern sector of the Colorado orogeny extends into?
Thought 1: I need to search Colorado orogeny, find the area that the eastern sector of the Colorado orogeny extends into, then find the elevation range of the area.
Action 1: Search[A figure of the regional map of Colorado orogeny, where eastern sector is displayed.]
Observation 1: Based on the give images, we can observe as follows. The Colorado orogeny was an episode of mountain building (an orogeny) in Colorado and surrounding areas. This took place from 1780 to 1650 million years ago (Mya), during the Paleoproterozoic (Statherian Period). It is recorded in the Colorado orogen, a >500-km-wide belt of oceanic arc rock that extends southward into New Mexico. The Colorado orogeny was likely part of the larger Yavapai progeny.
Thought 2: It does not mention the eastern sector of the Colorado orogeny. I need to look up eastern sector.
Action 2: Lookup[Extract the eastern sector information from the given figure.]
Observation 2: Based on the give images, we can observe as follows. The Colorado orogen, formerly called the Colorado province, is a >500-km-wide belt of oceanic arc rock (1.78–1.65 Ga) that extends southward into New Mexico and composes a major part of the Proterozoic provinces of southwestern United States. This transcontinental collisional event occurred during the Paleoproterozoic (Statherian Period). The Wyoming sector of the Colorado orogeny was formerly called the Medicine Bow orogeny. The eastern sector extends into the High Plains and is called the Central Plains orogeny. The boundary between the Colorado orogeny and the Wyoming craton is the Cheyenne belt, a 5-km-wide mylonitic shear zone that verges northward. The Cheyenne belt transects and cuts off the south edge of the older Trans-Hudson progeny.
Thought 3: The eastern sector of Colorado orogeny extends into the High Plains, so High Plains is the area. I need to search High Plains and find its elevation range.
Action 3: Search[A figure of High Plains, where its elevation range is annotated.]
Observation 3: Based on the give images, we can observe as follows. The High Plains are a subregion of the Great Plains, mainly in the Western United States, but also partly in the Midwest states of Nebraska, Kansas, and South Dakota, generally encompassing the western part of the Great Plains before the region reaches the Rocky Mountains. The High Plains are located in eastern Montana, southeastern Wyoming, southwestern South Dakota, western Nebraska, eastern Colorado, western Kansas, eastern New Mexico, western Oklahoma, and to just south of the Texas Panhandle. The southern region of the Western High Plains ecology region contains the geological formation known as Llano Estacado which can be seen from a short distance or on satellite maps. From east to west, the High Plains rise in elevation from around 1,800 to 7,000 ft (550 to 2,130 m).
Thought 4: High Plains rise in elevation from around 1,800 to 7,000 ft, so the answer is 1,800 to 7,000 ft.
Action 4: Finish[1,800 to 7,000 ft]

Example 2:

Question: Which magazine was started first Arthur's Magazine or First for Women?
Thought 1: I need to search Arthur's Magazine and First for Women, and find which was started first.
Action 1: Search[A wiki document page about Arthur's Magazine.]
Observation 1: Based on the give images, we can observe as follows. Arthur's Magazine (1844–1846) was an American literary periodical published in Philadelphia in the 19th century. Edited by Timothy Shay Arthur, it featured work by Edgar A. Poe, J.H. Ingraham, Sarah Josepha Hale, Thomas G. Spear, and others. In May 1846 it was merged into Godey's Lady's Book.
Thought 2: Arthur's Magazine was started in 1844. I need to search First for Women next.
Action 2: Search[A webpage introduction about First for Women.]
Observation 2: Based on the give images, we can observe as follows. First for Women is a woman's magazine published by Bauer Media Group in the USA. The magazine was started in 1989. It is based in Englewood Cliffs, New Jersey. In 2011 the circulation of the magazine was 1,310,696 copies.
Thought 3: First for Women was started in 1989. 1844 (Arthur's Magazine) < 1989 (First for Women), so Arthur's Magazine was started first.
Action 3: Finish[Arthur's Magazine]

Example 3:

Question: Guitars for Wounded Warriors is an album that was recorded in the village in which New York county?
Thought 1: I need to search Guitars for Wounded Warriors, then find in which New York county it was recorded.
Action 1: Search[A document page about Guitars for Wounded Warriors.]
Observation 1: Based on the give images, we can observe as follows. Guitars for Wounded Warriors is the sixth solo studio album by guitarist Brian Tarquin, released in August 2014 by Cleopatra Records. In 2016, it received a Best Compilation Album nomination by the Independent Music Awards. All tracks were recorded at Tarquin's Jungle Room Studios in New Paltz (village), New York. Being moved by the lack of support for returning veterans through his life Tarquin decided to compose a heavy instrumental rock album as a way to show his appreciation to all veterans. So once again he enlisted top notch players to guest with him on the album, Billy Sheehan on Black Hawk, Ron "Bumblefoot" Thal and Reb Beach on Taliban Terror, Steve Morse on Freedom, Chris Poland on Charlie Surfs and Hunting, Chuck Loeb on Escape Kabul, Hal Lindes on Sand & Blood, Gary Hoey on 5 Klicks To Hell and Baghdad, Alex De Rosso Dokken on Broken Arrow, and The Flyin' Ryan Brothers on Surgical Strike. The entire album was engineered, produced and composed by Tarquin especially for each one of the guest guitarists. Partial proceeds are donated to the Fisher House Foundation from sales.
Thought 2: Guitars for Wounded Warriors was recorded in New Paltz (village), New York. I need to search New Paltz (village), New York and find which county it is in.
Action 2: Search[A figure of map that indicates the location of New Paltz (village), New York.]
Observation 2: Based on the give images, we can observe as follows. New Paltz is a village in Ulster County located in the U.S. state of New York. It is approximately 80 miles (130 km) north of New York City and 70 miles (110 km) south of Albany. The population was 7,324 at the 2020 census.
Thought 3: New Paltz (village), New York is in Ulster County, so the answer is Ulster County.
Action 3: Finish[Ulster County]

ConECT Dataset: Overcoming Data Scarcity in Context-Aware E-Commerce MT

Mikołaj Pokrywka[1,2] **Wojciech Kusa**[1,3] **Mieszko Rutkowski**[1] **Mikołaj Koszowski**[1]
[1]Machine Learning Research Allegro.com [2]Laniqo.com [3]NASK
Correspondence: {name}.{surname}@allegro.com

Abstract

Neural Machine Translation (NMT) has improved translation by using Transformer-based models, but it still struggles with word ambiguity and context. This problem is especially important in domain-specific applications, which often have problems with unclear sentences or poor data quality. Our research explores how adding information to models can improve translations in the context of e-commerce data. To this end we create ConECT– a new Czech-to-Polish e-commerce product translation dataset coupled with images and product metadata consisting of 11,400 sentence pairs. We then investigate and compare different methods that are applicable to context-aware translation. We test a vision-language model (VLM), finding that visual context aids translation quality. Additionally, we explore the incorporation of contextual information into text-to-text models, such as the product's category path or image descriptions. The results of our study demonstrate that the incorporation of contextual information leads to an improvement in the quality of machine translation. We make the new dataset publicly available.[1]

1 Introduction

Neural Machine Translation (NMT) has significantly advanced the field of machine translation by leveraging Transformer-based models (Bahdanau et al., 2015; Vaswani et al., 2017b). These models have been critical in enhancing translation quality, particularly by incorporating mechanisms such as cross-attention to achieve better semantic understanding. However, despite these improvements, sentence-level translation in NMT often struggles with issues such as contextual disambiguation (Rios Gonzales et al., 2017). For example, the word *"pen"* can refer to a writing instrument or an enclosure for animals, depending on the context. These ambiguities present a significant challenge in achieving accurate translations based solely on the context of single sentences.

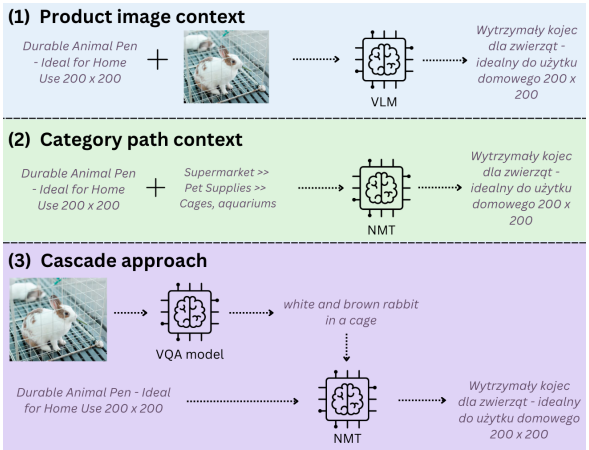


Figure 1: We evaluate three methods for contextualisation in e-commerce MT: (1) combining images with text in VLM, (2) appending category path context in NMT, and (3) a cascade approach consisting of a vision Q&A and a text-to-text NMT.

In an attempt to address these limitations, Multimodal Machine Translation (MMT) has emerged as a promising paradigm (Specia et al., 2016; Shen et al., 2024). MMT integrates visual information alongside textual data to provide additional context, thereby enhancing translation quality. Studies have shown that leveraging visual cues can significantly improve the disambiguation of lexical items and contribute to more accurate translations (Yao and Wan, 2020; Wang and Xiong, 2021). For instance, visual context can help resolve ambiguities in sentences where the textual information alone is insufficient (Liu et al., 2021). However, high-quality machine translation training data with aligned images is available only for a fraction of the parallel corpora.

In this work, we explore the effectiveness of integrating context information into a domain-specific translation task. We create a new testset for Czech-

[1] https://huggingface.co/datasets/allegro/ConECT

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 79–86
July 27 - August 1, 2025

to-Polish translation in e-commerce, and test three approaches for context-aware MT (Figure 1). We fine-tune and evaluate a VLM, as well as traditional NMT models for which we integrate contextual information as special instruction tokens. Our findings demonstrate that visual context can enhance translation quality in domain-specific scenarios, serving as a valuable additional feature.

Our contributions are as follows:

- We release a new multimodal dataset for e-commerce MT for the under-researched cs-pl language pair;
- We fine-tune and evaluate a VLM on the domain-specific MT task;
- We propose a method for encoding category paths and image descriptions in small NMT models.

We start by describing related work (§2), then introduce the ConECT dataset (§3), explain our experimental setup (§4), and finally discuss the results (§5).

2 Related work

In this section we discuss multimodal and e-commerce product-oriented MT.

2.1 Multimodal Machine Translation

Multimodal Machine Translation (MMT), which integrates visual context into machine translation, has garnered significant attention in recent years (Elliott et al., 2016; Shen et al., 2024). Research has shown that visuals can effectively bridge linguistic gaps between languages (Chen et al., 2019; Sigurdsson et al., 2020; Su et al., 2019). Early efforts by Calixto and Liu (2017) incorporated global image features into the encoder or decoder of NMT models. Subsequent studies have explored the use of more granular image contexts, such as spatial image regions with attention mechanisms (Calixto et al., 2017a; Caglayan et al., 2016).

Yang et al. (2020) introduced a method for joint training of source-to-target and target-to-source models to promote visual agreement. Yin et al. (2020) built a multi-modal graph linking image objects and source words, leveraging an external visual grounding model for alignment.

2.2 Benchmarks and Shared Tasks in MMT

Between 2016 and 2018, WMT organised a shared task on MMT (Specia et al., 2016; Elliott et al., 2017; Barrault et al., 2018). The organisers primarily used the Multi30K dataset (Elliott et al., 2016) and metrics like BLEU, METEOR, and TER. In the first edition of the shared task, Specia et al. (2016) found that neural MMT models initially underperformed compared to text-only SMT models. However, Elliott et al. (2017) and Barrault et al. (2018) expanded languages and test datasets, noting performance improvements in MMT models, especially when incorporating external resources.

Elliott (2018) introduced an adversarial evaluation method to assess whether multimodal translation systems effectively utilize visual context. This method evaluates the performance difference of a system when provided with either a congruent or an incongruent image as additional context.

Futeral et al. (2023) introduced the *CoMMuTE* (Contrastive Multilingual Multimodal Translation Evaluation) dataset to evaluate multimodal machine translation systems with a focus on resolving ambiguity using images. Their approach, which includes neural adapters and guided self-attention, showed significant improvement over text-only models, particularly in English–French, English–German, and English–Czech translations. Futeral et al. (2024) extended the setup with *ZeroMMT*, a technique for zero-shot multimodal machine translation that does not rely on fully supervised data. This method, which uses visually conditioned masked language modelling and Kullback-Leibler divergence training, demonstrated near state-of-the-art performance and was extended to Arabic, Russian, and Chinese.

2.3 E-commerce product-oriented MT

An e-commerce product-oriented machine translation task uses product images and product metadata as inputs. Calixto et al. (2017c) pioneered this task with a bilingual product description dataset, evaluating models such as phrase-based statistical MT (PBSMT), text-only MT, and MMT. Their results highlighted PBSMT's superior performance, with MMT models enhancing translation when reranking PBSMT outputs. Calixto et al. (2017b) highlight the potential impact of multi-modal NMT in the context of e-commerce product listings. With only a limited amount of multimodal and multilingual training data available, both text-only and

multi-modal NMT models failed to outperform a productive SMT system.

Song et al. (2021) introduced a large-scale dataset along with a unified pre-training and fine-tuning framework, proposing pre-training tasks for aligning bilingual texts and product images. These tasks include masked word reconstruction with bilingual and image context, semantic matching between text and image, and masking of source words conveying product attributes. This framework contributed to more robust translation models.

2.4 Visual information integration in LLMs

The field of NLP has evolved with the introduction of large language models (LLMs) (Ouyang et al., 2022; Achiam et al., 2023; Touvron et al., 2023). Since traditional text-only MT has advanced to LLM-based methods (Xu et al., 2023), leveraging LLMs for MMT is a promising direction. Current multi-modal LLMs, whether using linear models (Liu et al., 2024) or Query Transformers (Li et al., 2023), often suffer from visual information loss. Enhancing alignment between textual and visual modalities and adaptively extracting relevant visual information are critical for optimizing LLM performance in MMT tasks.

3 ConECT dataset

The ConECT dataset (**Con**textual **E**-**C**ommerce **T**ranslation) is designed to support research on context-aware MT in the e-commerce domain. To create this dataset, we extracted 11,000 sentences in Polish from the `allegro.pl` e-commerce platform. Next, we aligned a primary product image and category path. The sentences were then manually translated into Czech by professional translators. Each translation was reviewed to ensure accuracy and contextual relevance. A detailed breakdown of the dataset statistics is provided in Table 1.

The ConECT dataset is divided into various content types to provide comprehensive coverage of e-commerce translation contexts, as summarized below.

Product names Product names are typically short phrases that precisely identify a product. They often contain specific terminology, brand names, and product specifications.

Product descriptions Product descriptions are longer texts that provide detailed information about a product, including its features, specifications, usage instructions, and benefits. These texts can be more descriptive and less structured than product names.

Split	Content type	#Sent.	#Img.	#category paths	CS #Tokens	CS Len.	CS Vocab
Test	Offer titles	1,924	1,920	840	13,898,	7.2	6,206
	Prod. desc.	3,680	2,905	1,090	38,885,	10.6	13,286
	Prod. names	4,691	4,659	1,361	34,886,	7.4	10,422
	ALL	10,295	6,146	1,542	87,669,	8.5	22,121
Valid	Offer titles	203	203	165	1,449	7.1	1,080
	Prod. desc.	403	389	285	4,118	10.2	2,472
	Prod names	505	505	360	3,757	7.4	2,116
	ALL	1,111	1,042	596	9,324	8.4	4,822

Table 1: ConECT dataset statistics. *Len.* denotes average length in words. Polish sentences have similar statistics.

Offer titles Offer titles are concise and attractive phrases crafted by marketers to engage potential buyers, often including promotional language, discounts, or special offers. This category can be challenging due to the need to maintain both the persuasive tone and the specific promotional content during the translation process.

4 Experimental setup

In this section we describe the training and evaluation data, the models used, and the evaluation procedure.

4.1 Training data

To create a parallel e-commerce dataset, we aligned Polish and Czech product names and descriptions for corresponding products that were listed on both e-commerce platforms `allegro.pl` and `mall.cz`. Merchants manually translated original Czech product names and descriptions into Polish. To create sentence-level pairs of multi-sentence descriptions, we split them into sentences and aligned them using a language-agnostic BERT sentence embedding model (Feng et al., 2022). We compared every sentence in one description with every sentence in the corresponding description in the other language to align the sentence pairs. That procedure resulted in a dataset containing 230,000 parallel sentences, each paired with product category paths and one of 38,000 unique images.

Data with image context For experiments involving translation with image context, we additionally collected 440,000 Polish product names paired with 430,000 unique images from the `allegro.pl` e-commerce platform. These product names were back-translated into Czech, creating a synthetic dataset of product names with image

context. The images were in JPEG format with varying sizes and were resized to 224x224 pixels.

Text-to-text models For the baseline text-to-text models, we used 53 million sentence pairs, primarily drawn from the OPUS corpora (Tiedemann and Nygaard, 2004) and internal e-commerce domain data.

For fine-tuning with category paths as context, we extended the 230,000 parallel sentences from the original dataset with 7 million back-translated product names and 7 million back-translated product description sentences, paired with their respective category paths. Additionally, we extended the fine-tuning dataset by incorporating 7 million parallel sentences without category paths from the baseline model's training set. The category paths were represented as text, listing the hierarchical subcategories for each product (e.g., "*Sports* » *Bicycles* » *Tires*").

For experiments involving image descriptions, we generated image descriptions in Czech using the `paligemma-3b-mix-224`[2] (Beyer et al., 2024) model on all of the previously mentioned data with image context. Additionally, for fine-tuning, we included 700,000 sentences without image descriptions, extracted from the baseline model's training set.

4.2 Models

Models with image context Experiments involving translation with image context were conducted on the `paligemma-3b-pt-224`[3] model. Our primary goal was to evaluate the influence of images on translation. To achieve this, we fine-tuned the models for the translation task with two types of image data: (1) original corresponding product images, and (2) a black image unrelated to the text input. Further details of the experimental setup are given in Appendix A.1.

Text-to-text baseline model We developed a sentence-level text-to-text baseline model using the Transformer (big) architecture (Vaswani et al., 2017a), trained with the Marian framework (Junczys-Dowmunt et al., 2018). Details of the experimental setup are given in Appendix A.2.

Models with category context To ensure a fair comparison, we implemented two fine-tuning approaches on the baseline model: one that incorporates category context and one that does not. The category context was integrated by adding a prefix containing the product's category path in Polish to the source sentences. The category path was enclosed with special tokens <SC> and <EC> to separate it clearly from the source sentence. The model without category context was trained using the same configuration and data setup, but without the category path prefixes. Details of the experimental setup are given in Appendix A.3.

Models with image descriptions Following a similar approach as in the category context experiments, we fine-tuned the baseline model with and without prefixes containing image descriptions. The image descriptions were added as prefixes wrapped with <SD> and <ED> tokens. This model was trained exclusively on data with image context converted into image descriptions and did not include any data from category context experiments. Details of the experimental setup are given in Appendix A.4.

NLLB-baseline For comparison, we report our results alongside the NLLB-200-Distilled-600M model.[4]

Evaluation metrics We use sacreBLEU (Post, 2018) to calculate the chrF[5] (Popović, 2015) score, and the Unbabel/wmt22-comet-da[6] (Rei et al., 2022) model to calculate the COMET metric. We used sacreCOMET (Zouhar et al., 2024) to create the COMET setup signature.

5 Results and discussion

A performance comparison of models on `ConECT` test sets is presented in Table 2.

Models with image context For the PaliGemma models, those fine-tuned and evaluated with appropriate images outperformed those with unrelated images. Notable improvements were observed in the product names and offer titles sets. However, while the COMET metric showed a decrease for the product descriptions, the chrF metric showed a slight increase.

Models with category context Both fine-tuned models showed improved performance, with the

[2] https://huggingface.co/google/paligemma-3b-mix-224

[3] https://huggingface.co/google/paligemma-3b-pt-224

[4] https://huggingface.co/facebook/nllb-200-distilled-600M

[5] chrF signature: nrefs:1|case:mixed|eff:yes|nc:6|nw:0|space:no|version:2.3.1

[6] Python3.9.19|Comet2.2.2|fp32|Unbabel/wmt22-comet-da

Model	Train	Inference	Product names		Offer titles		Product desc		All sets	
			chrF	COMET	chrF	COMET	chrF	COMET	chrF	COMET
NLLB-600M	–	–	48.46	0.7214	38.01	0.6537	48.50	0.7774	46.85	0.7288
PaliGemma-3b	real img	real img	**83.48**	**0.9310**	**79.41**	**0.9083**	**61.92**	0.8987	**72.31**	**0.9152**
	real img	black img	81.36	0.9224	77.10	0.8972	61.75	**0.8994**	71.12	0.9095
	black img	real img	82.69	0.9275	77.86	0.9009	60.57	0.8891	71.15	0.9088
	black img	black img	82.49	0.9268	77.97	0.9009	60.87	0.8908	71.24	0.9091
Baseline	–	–	84.83	0.9326	83.73	0.9227	70.76	0.9335	77.74	0.9311
Category paths experiements	no category context		85.27	0.9372	83.66	0.9242	**72.78**	0.9389	**78.87**	0.9354
	category context		**85.51**	**0.9385**	**83.73**	**0.9248**	71.95	**0.9393**	78.56	**0.9362**
Image desc. experiments	no description context		**85.10**	**0.9367**	**83.99**	**0.9246**	**70.81**	**0.9358**	**77.90**	**0.9341**
	description context		83.25	0.8673	82.63	0.8974	48.26	0.7243	65.97	0.8219

Table 2: Comparison of the results on the ConECT test set shows that the VLM model with image context and the NMT model with category paths achieved improved performance due to the added context. However, experiments with synthetic image descriptions led to a decrease in metrics.

version incorporating category path context achieving a notable advantage in the COMET metric across all datasets. The most significant improvement was observed with product names, while the smallest gain in the COMET metric occurred with product descriptions, where the chrF metric actually decreased.

Models with image descriptions The model using image description prefixes showed a significant decrease in quality, especially on the product description dataset. It is important to note that the fine-tuning was performed on synthetic image descriptions and used a smaller dataset than the models incorporating category context. In this case, the added context had a negative impact on performance, highlighting that fine-tuning an NMT model with prefixes can degrade its quality in some scenarios.

6 Conclusion

This study explores methods for incorporating context into MT using the ConECT dataset. We are making this dataset publicly available to support research into context-aware translation tasks. We investigated the fine-tuning of VLM for machine translation to exploit image-based context for improved translation quality. Secondly, we analysed the effect of product category paths on translation performance in text-to-text models for e-commerce data. Both experiments showed that the models benefited from contextual information. We also report negative results from fine-tuning with image description prefixes, highlighting that added context can sometimes impair model quality and that this straightforward approach requires further refinement.

Limitations

Our approaches rely heavily on the quality of training data and the suitability of the test set for context-aware translation. In many cases, the text alone is sufficient without additional context. Moreover, incorporating extra context can sometimes reduce translation quality, especially with LLMs, where hallucinations may introduce critical errors for users. The experiments with VLM discussed in this paper require significantly more computational resources than text-to-text NMT models, due to the larger model and data sizes. We used AI assistance exclusively to enhance the text style and identify grammatical errors in this manuscript.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*.

Loïc Barrault, Fethi Bougares, Lucia Specia, Chiraag Lala, Desmond Elliott, and Stella Frank. 2018. Findings of the third shared task on multimodal machine translation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 304–323, Belgium, Brussels. Association for Computational Linguistics.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. 2024. Paligemma: A versatile 3b vlm for transfer. *Preprint*, arXiv:2407.07726.

Ozan Caglayan, Loïc Barrault, and Fethi Bougares. 2016. Multimodal attention for neural machine translation. *arXiv preprint arXiv:1609.03976*.

Iacer Calixto and Qun Liu. 2017. Incorporating global visual features into attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 992–1003, Copenhagen, Denmark. Association for Computational Linguistics.

Iacer Calixto, Qun Liu, and Nick Campbell. 2017a. Doubly-attentive decoder for multi-modal neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1913–1924, Vancouver, Canada. Association for Computational Linguistics.

Iacer Calixto, Daniel Stein, Evgeny Matusov, Sheila Castilho, and Andy Way. 2017b. Human evaluation of multi-modal neural machine translation: A case-study on E-commerce listing titles. In *Proceedings of the Sixth Workshop on Vision and Language*, pages 31–37, Valencia, Spain. Association for Computational Linguistics.

Iacer Calixto, Daniel Stein, Evgeny Matusov, Pintu Lohar, Sheila Castilho, and Andy Way. 2017c. Using images to improve machine-translating E-commerce product listings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 637–643, Valencia, Spain. Association for Computational Linguistics.

Shizhe Chen, Qin Jin, and Jianlong Fu. 2019. From words to sentences: A progressive learning approach for zero-resource machine translation with visual pivots. *arXiv preprint arXiv:1906.00872*.

Desmond Elliott. 2018. Adversarial evaluation of multimodal machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2974–2978, Brussels, Belgium. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation*, pages 215–233, Copenhagen, Denmark. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. *Preprint*, arXiv:2007.01852.

Matthieu Futeral, Cordelia Schmid, Ivan Laptev, Benoît Sagot, and Rachel Bawden. 2023. Tackling ambiguity with images: Improved multimodal machine translation and contrastive evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5394–5413, Toronto, Canada. Association for Computational Linguistics.

Matthieu Futeral, Cordelia Schmid, Benoît Sagot, and Rachel Bawden. 2024. Towards zero-shot multimodal machine translation. *arXiv preprint arXiv:2407.13579*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Pengbo Liu, Hailong Cao, and Tiejun Zhao. 2021. Gumbel-attention for multi-modal machine translation. *arXiv preprint arXiv:2103.08862*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Annette Rios Gonzales, Laura Mascarell, and Rico Sennrich. 2017. Improving word sense disambiguation in neural machine translation with sense embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 11–19, Copenhagen, Denmark. Association for Computational Linguistics.

Huangjun Shen, Liangying Shao, Wenbo Li, Zhibin Lan, Zhanyu Liu, and Jinsong Su. 2024. A survey on multi-modal machine translation: Tasks, methods and challenges. *arXiv preprint arXiv:2405.12669*.

Gunnar A Sigurdsson, Jean-Baptiste Alayrac, Aida Nematzadeh, Lucas Smaira, Mateusz Malinowski, Joao Carreira, Phil Blunsom, and Andrew Zisserman. 2020. Visual grounding in video for unsupervised word translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10850–10859.

Yuqing Song, Shizhe Chen, Qin Jin, Wei Luo, Jun Xie, and Fei Huang. 2021. Product-oriented machine translation with cross-modal cross-lingual pretraining. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2843–2852.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 543–553, Berlin, Germany. Association for Computational Linguistics.

Yuanhang Su, Kai Fan, Nguyen Bach, C-C Jay Kuo, and Fei Huang. 2019. Unsupervised multi-modal neural machine translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10482–10491.

Jörg Tiedemann and Lars Nygaard. 2004. The OPUS corpus - parallel and free: `http://logos.uio.no/opus`. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Dexin Wang and Deyi Xiong. 2021. Efficient object-level visual context modeling for multimodal machine translation: Masking irrelevant objects helps grounding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 2720–2728.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2023. A paradigm shift in machine translation: Boosting translation performance of large language models. *arXiv preprint arXiv:2309.11674*.

Pengcheng Yang, Boxing Chen, Pei Zhang, and Xu Sun. 2020. Visual agreement regularized training for multi-modal machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9418–9425.

Shaowei Yao and Xiaojun Wan. 2020. Multimodal transformer for multimodal machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4346–4350, Online. Association for Computational Linguistics.

Yongjing Yin, Fandong Meng, Jinsong Su, Chulun Zhou, Zhengyuan Yang, Jie Zhou, and Jiebo Luo. 2020. A novel graph-based multi-modal fusion encoder for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3035, Online. Association for Computational Linguistics.

Vilém Zouhar, Pinzhen Chen, Tsz Kin Lam, Nikita Moghe, and Barry Haddow. 2024. Pitfalls and outlooks in using comet. *Preprint*, arXiv:2408.15366.

A Details on experiments

We conducted all our experiments on a single server equipped with four Nvidia A100 GPUs, each with 80 GB of RAM.

A.1 VLM setup

PaliGemma was fine-tuned using LoRA (Hu et al., 2021) with rank $r = 8$ and alpha $\alpha = 8$. The fine-tuning was performed on a single A100 GPU for 4 epochs with a learning rate of $1e^{-4}$ and batch size set to 16.

A.2 Text-to-text baseline model

The model was trained on four NVIDIA A100 GPUs. It employs a shared vocabulary of 32,000 subword tokens, generated using the SentencePiece toolkit (Kudo and Richardson, 2018), with all embeddings tied during training. Early stopping was set to 10, with the validation frequency set to 3000 steps and based on the chrF metric on the `ConECT` validation set.

A.3 Models with category context

For fine-tuning with prefixes we employed the following special tokens to mark category context in the SentencePiece vocabulary:

- <SC> – start of the category path
- <SEP> – separator of subcategories
- <EC> – end of the category path

The subcategories were provided in the target language and were not included as special tokens in the vocabulary. An example of a source sentence is as follows: *<SC> Moda <SEP> Odzież, Obuwie, Dodatki <SEP> Obuwie <SEP> Męskie <SEP> Sportowe <EC> Big Star pánské sportovní boty JJ174278 černé 44*. The target sentences remained unchanged.

The model without category context was trained using the same configuration and data setup, but without the category path prefixes. Both fine-tunings were performed on two NVIDIA A100 GPUs with a learning rate of $5e^{-6}$, and early stopping was applied based on the chrF metric on the concatenated `ConECT` validation set.

A.4 Models with image descriptions

The image descriptions were generated in the Czech and Polish languages using the `google/paligemma-3b-mix-224` model and the `transformers` library. However, only image descriptions in Czech were used in the experiments. The prompts were structured as simple tasks as found in the original PaliGemma paper (Beyer et al., 2024). The exact prompts are shown in Table 3. Images for inference were resized to 224x224 pixels. We include generated Czech and Polish captions in the dataset. For fine-tuning with image description we employed the following special tokens in the SentencePiece vocabulary:

- <SD> – start of the image description
- <ED> – end of the image description

An example of a source sentence is as follows: *<SD> Černé a bílé boty s nápisem " big star " na boku. <ED> Big Star pánské sportovní boty JJ174278 černé 44*

The training configuration was identical to the experiments with category context, except that the validation frequency was reduced to every 300 steps.

Target language	Prompt used for image description
Czech	`popsat obrázek v češtině`
Polish	`opisz obrazek po polsku`

Table 3: Prompts for image description generation used with the paligemma-3b-mix-224 model.

A Measure of the System Dependence of Automated Metrics

Pius von Däniken and **Jan Milan Deriu** and **Mark Cieliebak**
Centre for Artificial Intelligence
ZHAW School of Engineering
{vode,deri,ciel}@zhaw.ch

Abstract

Automated metrics for Machine Translation have made significant progress, with the goal of replacing expensive and time-consuming human evaluations. These metrics are typically assessed by their correlation with human judgments, which captures the monotonic relationship between human and metric scores. However, we argue that it is equally important to ensure that metrics treat all systems fairly and consistently. In this paper, we introduce a method to evaluate this aspect.

1 Introduction

Recent years have seen significant advances in machine translation (MT), marked notably by the introduction of the transformer architecture (Vaswani et al., 2017). Current large-scale commercial systems such as GPT (Brown et al., 2020) continue this trend and show promising results (Kocmi et al., 2023; Hendy et al., 2023; Wu and Hu, 2023). A critical supplement to these advancements is thorough and reliable evaluation procedures, which are essential not only for measuring overall progress but also for effectively comparing different systems. While evaluation based on human raters is still considered the gold standard, it is expensive and time-intensive. Therefore, considerable efforts have been made to develop automated metrics for assessing translation quality. Notably, the WMT Metrics series of shared tasks are dedicated to this purpose (Freitag et al., 2023, 2022, 2021, i.a.). Automated metrics usually assign a scalar [1] quality rating to a candidate translation based on the source segment and a reference translation. A system-level rating is derived by averaging the segment ratings over a test set.

To measure a metric's usefulness, we usually measure two aspects: its correlation to human judgments on the segment-level (which checks if there is a monotonic function between metric ratings and human ratings) and whether the system-level ratings can reproduce the same ranking as human ratings (Kocmi et al., 2021; von Däniken et al., 2024). In this paper, we argue that this evaluation of metrics is insufficient, as it ignores a central requirement, namely, that it should treat all systems under evaluation equally. As stated more colloquially, a measuring stick should not change length depending on the measured object. However, this is exactly what we observe in current metrics.

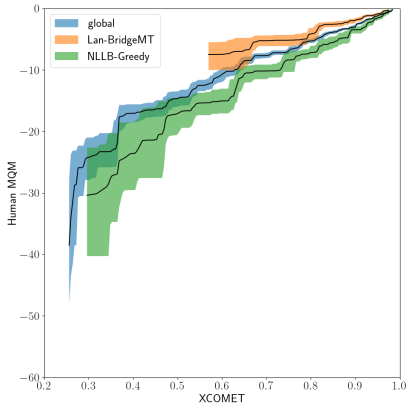


Figure 1: Average Human Ratings associated with *XCOMET* scores on Chinese to English (*zh-en*) WMT 23 data. We show scores for all system in aggregate (global) and two individual systems.

Consider Figure 1, which shows the expected human rating for each score of the *XCOMET* metric (the best metric in the WMT23 metrics task, with a very high segment-level correlation of 0.65 for the zh-en language pair) (Freitag et al., 2023). That is, for each possible value that *XCOMET* may assume, we show the expected human rating and the 95% confidence interval (computed using Isotonic Regression and bootstrap sampling; see Sections 2 and 3 for the details). The global curve (blue) shows the average human score for each metric score if computed over all systems under evalua-

[1] Other types of ratings exist, in particular preference ratings (Belz and Kow, 2010).

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 87–99
July 27 - August 1, 2025

tion in the WMT23 dataset (in standard correlation to human judgment evaluation, we only measure whether this curve is monotonic). In contrast, *Lan-BridgeMT* (best system according to humans) and *NLB-Greedy* (lowest-rated system according to humans) show the average human score for each metric score when computed on one separate system only. For instance, an *XCOMET* score of 0.7 corresponds to an average Human-MQM score of -5.2 for *Lan-BridgeMT*, and Human-MQM score of -10.2 for *NLB-Greedy*.

This leads to the following consequence: For *Lan-BridgeMT*, higher human scores are associated with lower metric scores than the global curve, which leads to an underestimation of *Lan-BridgeMT*'s performance, according to *XCOMET*. The opposite effect is visible for *NLB-Greedy*, which is overestimated and, in fact, gains 3 ranks (from 15th to 12th place) when comparing the metric and human ranking (see also Table 1 in Section 3). Thus, a metric that exhibits a high global segment-level correlation to human judgments can lead to wrong system-level rankings. This observation leads us to the central claim of this paper: **The cause of the discrepancy between the correlation on the segment level and the final system ranking is due to the metric's dependency of the system under evaluation.**

The main position of this paper is that when evaluating a novel metric, one ought to measure the dependency on the system under evaluation as well, alongside the correlation to human judgment. In the following, we will formalize this dependency of the relation between human and metric ratings on the system under evaluation and derive a measure for quantifying this effect.

2 Averaging Metric Scores

Assume we are given a set of K machine translation systems π_k to evaluate. A translation system maps an input sentence $i \in \mathcal{I}$ in a fixed source language to an output sentence $o \in \mathcal{O}$ in a fixed target language: $\pi_k : \mathcal{I} \rightarrow \mathcal{O}$. The usual human evaluation scenario involves curating a test set of N inputs $\mathcal{T} = \left\{i^{(j)} | 1 \leq j \leq N\right\} \subset \mathcal{I}$ for which we collect the output of each system π_k for each input $i \in \mathcal{T}$, and then ask human annotators to produce ratings. This results in a set of ratings $\left\{(h_1^{(j)}, \ldots, h_k^{(j)}, \ldots, h_K^{(j)}) | 1 \leq j \leq N\right\}$, where $h_k^{(j)} \in \mathbb{R}$ is a scalar rating provided by human annotators measuring the quality of the translation provided by π_k for input $i^{(j)}$. We will assume that higher human ratings indicate higher translation quality. In this setting, it is natural to measure the overall quality of system π_k by the average human rating it achieves $\hat{\mu}_k^H = \frac{1}{N}\sum_{j=1}^{N} h_k^{(j)}$. This is an estimator of the expected human rating $\mu_k^H = \mathbb{E}[h_k]$ achieved by π_k for any input in $\mathcal{I}$, assuming that $\mathcal{T}$ is appropriately chosen.

In many cases, we want to replace human raters with an automated scalar metric $M : \mathcal{I} \times \mathcal{O} \rightarrow \mathbb{R}$, which maps an input and translation to a scalar value. For our test set $\mathcal{T}$, we can collect all metric ratings $\left\{(m_1^{(j)}, \ldots, m_k^{(j)}, \ldots, m_K^{(j)}) | 1 \leq j \leq N\right\}$, where $m_k^{(j)} = M(i^{(j)}, \pi_k(i^{(j)}))$, the metric rating for input $i^{(j)}$ and translation by π_k. In this case, it is common to use the sample average $\hat{\mu}_k^M = \frac{1}{N}\sum_{j=1}^{N} m_k^{(j)}$ to measure the quality of system π_k, which is an estimator of the expected metric rating $\mu_k^M = \mathbb{E}[m_k]$ achieved by π_k.

The goal of automated evaluation is to use $\hat{\mu}_k^M$ as a proxy measure for μ_k^H, in particular, to rank the systems $\pi_1, \ldots, \pi_K$ according to their performance. In the following, we will study the relationship between μ_k^H and μ_k^M, which is expressed by an unknown function f_G that maps from the metric scale to the human scale. There are two requirements to this function: first, that it is monotonic (i.e., that it respects the order of the metric scale), and second, that it does not depend on the system under evaluation π_k (i.e., that it is the same for all systems) The goal is to find the relation between μ_k^H and μ_k^M. The idea is to express $\mathbb{E}[h_k]$ in terms of an expectation over metric ratings as follows (for full derivation, see Appendix A):

$$\mathbb{E}[h_k] = \mathbb{E}_{p_k(m)}[\mathbb{E}_{p_k(h)}[h|m]] \tag{1}$$

The crucial element of Equation 1 is the conditional expectation $\mathbb{E}_{p_k(h)}[h|m]$. Here we consider the expectation according to $p_k(h)$, the distribution of human ratings for system π_k. Equation 1 describes the relationship between μ_k^H and μ_k^M by expressing the expected human rating in terms of an expectation over metric ratings. We interpret this element as a function f_k, which takes a metric rating as input and returns the expected human rating. Equation 1 yields a function f_k for each system separately, which is not necessarily the same across systems. At this point, we can restate

the introductory discussion using our formalism. When averaging metrics $\hat{\mu}_k^M$ to rank systems, we implicitly assume that there is a global function f_G that is equal to all the system-specific functions f_k, i.e., $f_G = f_1 = \cdots = f_K$, and thus, only measure if f_G is monotonic (through correlation to human judgments). However, as shown in Figure 1, this assumption does not hold in practice (where blue is f_G, and we have an f_k for the two other systems respectively). To show that this is insufficient, we consider the effects of violating the assumption. Let us assume $f_1 \neq f_2$, but both are monotonic. Consider the extreme example that $\mu_1^M = \mu_2^M$, i.e., systems π_1 and π_2 are of the same quality under the metric. However, consider the case $f_1(m) = f_2(m) + C$, $C > 0$. Then $\frac{1}{N}\sum_j f_1(m_1^{(j)}) = C + \frac{1}{N}\sum_j f_2(m_1^{(j)}) > \frac{1}{N}\sum_j f_2(m_2^{(j)})$, thus, yielding that π_1 is better than π_2 in human space. This shows the necessity of measuring the monotonicity of a global function f_G and the dependence of the metric on the systems under evaluation.

We first introduce the Expected Deviation (ED), which measures the difference between f_G and f_k for all $k \in \{1 \ldots K\}$, which tells us how much a system is over-or-underestimated according to the metric. That is the difference

$$ED(k) = \frac{1}{N}\sum_{j=1}^{N} f_G(m_k^{(j)}) - \frac{1}{N}\sum_{j=1}^{N} f_k(m_k^{(j)}) \quad (2)$$

This is equivalent to $\mu_k^G - \mu_k^H$, where $\mu_k^G = \frac{1}{N}\sum_{j=1}^{N} f_G(m_k^{(j)})$, thus, we measure the difference between the average rating according to the global function and the average rating of the system-specific function, which corresponds to the human rating-average. Note that a mis-ranking occurs if one system is severely overrated while another is severely underrated. Thus we define the system dependence score *SysDep*$(\mathcal{M})$ as the worst case of this effect:

$$SysDep(\mathcal{M}) = max_{\pi_k} ED(k) - min_{\pi_k} ED(k) \quad (3)$$

3 Experiments

Estimating the Conditional Expectation. Even though the functions f_k and f_G are unknown in general, we can estimate them from data. We will use *Isotonic Regression (IR)* (Barlow and Brunk, 1972) for this purpose, which estimates a monotonic function $\hat{f}_k$ minimizing $\sum_j(\hat{f}_k(m_k^{(j)}) - h_k^{(j)})^2$. To estimate f_G, we utilize the same approach, pooling all paired data from all systems. To compute the *SysDep* of a metric, we compute the ED for each MT system under that metric. For this, we compute the average human rating $\hat{\mu}_k^H = \frac{1}{N_H}\sum_{j=1}^{N_H} h_k^{(j)}$, the average metric rating $\hat{\mu}_k^M = \frac{1}{N_M}\sum_{j=1}^{N_M} m_k^{(j)}$, as well as average remapped rating $\hat{\mu}_k^G = \frac{1}{N_M}\sum_{j=1}^{N_M} \hat{f}_G(m_k^{(j)})$ for each MT system. We provide our code in Appendix E.

Data. We rely on data from the WMT 23 Metrics shared task (Freitag et al., 2023). The data includes translations for 3 language pairs: English to German (*en-de*), Hebrew to English (*he-en*), and Chinese to English (*zh-en*). The translations were produces by 12-15 systems (depending on the language pair) which participated in the general MT task (Kocmi et al., 2023). Human ratings are available in the form of MQM annotations (Lommel et al., 2014), which are based on error-span annotations by experts that are subsequently transformed into a numeric value by assigning scores to errors based on their severity. Here, we will present results for the *XCOMET* (Guerreiro et al., 2023) metric (best metric according to correlation to human judgments) and the *zh-en* language pair, where we have access to $N_M = 1976$ segments per system rated by the metric and $N_H = 1177$ of these segments rated with human MQM ratings. Results for the other language pairs and an additional metric are shown in Appendix B. To estimate the conditional expectation functions f_k, we use the 1177 paired ratings for each system π_k. We employ $B = 200$ bootstrap samples of the paired data to fit B IR models. Our estimate, $\hat{f}_k$, represents the average of these B IR models. In Figure 1, we also present the range between the 2.5% and 97.5% percentiles.

Results. We show the results in Table 1. We can see that the ED ranges from -0.82 to 1.996, thus yielding a *SysDep* score of 2.816. We see that both *Lan-BridgeMT* and *GPT4-5shot* are underrated by the metric (negative *ED*), but *Lan-BridgeMT* more so, enough to invert their order. At the bottom of the ranking, we see a relatively large absolute *ED*. Ranking errors reflect an interplay between the systems' rating gap and the *ED*s. For example, *Online-A* loses 2 ranks according to the metric even though it has the lowest absolute *ED*. We also note that even though $\hat{f}_G$ is monotonic, the ranking between the metric and the remapped scores does not match completely. This can be attributed to

	Human		Metric		Remapped		Exp. Deviation
	$\hat{\mu}_k^H$	R	$\hat{\mu}_k^M$	R	$\hat{\mu}_k^G$	R	ED
Lan-BridgeMT	-2.100	1	0.889	2	-2.920	2	*-0.820*
GPT4-5shot	-2.305	2	0.893	1	-2.800	1	-0.494
Yishu	-3.231	3	0.880	4	-3.179	4	0.052
ONLINE-B	-3.385	4	0.879	5	-3.188	5	0.197
HW-TSC	-3.398	5	0.883	3	-3.080	3	0.318
ONLINE-A	-3.785	6	0.856	8	-3.812	8	-0.027
ONLINE-Y	-3.792	7	0.868	6	-3.479	6	0.313
ONLINE-G	-3.857	8	0.864	7	-3.607	7	0.250
ONLINE-W	-4.062	9	0.848	9	-4.165	10	-0.103
ZengHuiMT	-4.232	10	0.846	10	-4.140	9	0.092
IOL-Research	-4.586	11	0.843	11	-4.251	11	0.335
ONLINE-M	-5.433	12	0.820	15	-4.907	15	0.526
ANVITA	-6.078	13	0.830	13	-4.602	13	1.475
NLLB-MBR-BLEU	-6.360	14	0.825	14	-4.726	14	1.634
NLLB-Greedy	-6.574	15	0.831	12	-4.578	12	**1.996**

Table 1: System rankings and average rating of WMT 23 *zh-en* systems according to *XCOMET*. The lowest score is in italics, and the highest is in bold.

the uncertainty introduced by bootstrapping and extrapolating to the unpaired metric ratings. It can be seen for *ONLINE-W* and *ZengHuiMT*, which have similar metric ratings.

Overall, our results show that although there is a highly monotonic function between the *XCOMET* scale and the human scale, *XCOMET* exhibits a high dependency on the system under evaluation, thus yielding an inconsistent ranking between humans and *XCOMET*.

4 Related Work

The derivation in Section 2 closely follows Wu and Resnick (2024), who provide the same argument in the context of binary prevalence estimation. In our case, the conditional expectation $\mathbb{E}[h|m]$ plays the same role as the calibration curve in their framework. Under that lens, the Expected Deviation is analogous to the Expected Calibration Error (Posocco and Bonnefoy, 2021). Following the same analogy, evaluating a new MT system is similar to applying a classifier to a new domain.

Previous studies by Deriu et al. (2023) and von Däniken et al. (2022) have highlighted that metric performance depends on the system under test. They employed a Bayesian framework to determine the proportions of binary or preference human ratings from metric scores; critically relying on confusion matrices estimated for each MT system. In this discrete rating context, these confusion matrices represent the same concept as $\mathbb{E}[h|m]$. In follow-up work, von Däniken et al. (2024) find that some metrics disproportionately favor certain MT systems over others compared to human preference ratings. Our finding provides a plausible explanation.

Chaganty et al. (2018) shows how to combine human ratings and metric ratings to derive an unbiased estimate of the true expected human rating μ^H while reducing the number of annotations needed. The proposed control variates estimator is based only on human and metric scores for a given MT system, even when estimating their correlation, thus avoiding the problem we describe.

Wei and Jia (2021) consider disagreements in the ordering of systems when using μ_k^M instead of μ_k^H. In particular they study the sign error, cases where $sign(\mu_1^M - \mu_2^M) \neq sign(\mu_1^H - \mu_2^H)$. They apply a bias variance decomposition to this error and find that while the human estimator is unbiased, it exhibits high variance while the opposite is the case for metrics. Our *SysDep* score presents a way to quantify this bias.

5 Conclusion

In this paper, we emphasize the importance of ensuring that automated metrics treat all MT systems consistently, a factor overlooked in current evaluations. By mapping metric scores to the human rating scale, we estimate how much a metric misjudges individual system performance. We compute the range of these deviations to assess how consistently a metric treats different systems. In Appendix C, we re-evaluate WMT23 metrics from this perspective. Additionally, in Appendix D, we confirm that these results stem from systematic differences in how metrics treat systems by measuring deviations within splits of a single system's ratings.

Limitations

This paper is intended to explore a supplementary aspect of the evaluation of automated metrics. The *SysDep* measure we developed will hopefully provide a starting point for the development of more refined evaluation of the way metrics treat different systems differently.

Our experiments are based solely on data from the WMT23 Metrics shared task. To further solidify our findings a larger scale study with more domains and larger sample sizes are needed.

While we provide a way to measure the system dependence of a metric, we do not provide any suggestions on how to develop metrics that minimize the *SysDep*.

Acknowledgments

This work was supported by the Swiss National Science Foundation (SNF) within the project "Unified Model for Evaluation of Text Generation Systems (UniVal)" [200020_219819].

References

R. E. Barlow and H. D. Brunk. 1972. The isotonic regression problem and its dual. *Journal of the American Statistical Association*, 67(337):140–147.

Anja Belz and Eric Kow. 2010. Comparing rating scales and preference judgements in language evaluation. In *Proceedings of the 6th International Natural Language Generation Conference*. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Arun Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Melbourne, Australia. Association for Computational Linguistics.

Jan Deriu, Pius von Däniken, Don Tuggener, and Mark Cieliebak. 2023. Correction of errors in preference ratings from automated metrics for text generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6456–6474, Toronto, Canada. Association for Computational Linguistics.

Markus Freitag, Nitika Mathur, Chi-kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Tom Kocmi, Frederic Blain, Daniel Deutsch, Craig Stewart, Chrysoula Zerva, Sheila Castilho, Alon Lavie, and George Foster. 2023. Results of WMT23 metrics shared task: Metrics might be guilty but references are not innocent. In *Proceedings of the Eighth Conference on Machine Translation*, pages 578–628, Singapore. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2023. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Preprint*, arXiv:2310.10482.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation. *Preprint*, arXiv:2302.09210.

Juraj Juraska, Mara Finkelstein, Daniel Deutsch, Aditya Siddhant, Mehdi Mirzazadeh, and Markus Freitag. 2023. MetricX-23: The Google submission to the WMT 2023 metrics shared task. In *Proceedings*

of the Eighth Conference on Machine Translation, pages 756–767, Singapore. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775, Singapore. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Arle. Language Technology Lab) Lommel, Hans. Language Technology Lab) Uszkoreit, and Aljoscha. Language Technology Lab) Burchardt. 2014. Multidimensional quality metrics (mqm) : a framework for declaring and describing translation quality metrics. *Tradumàtica*, (12):455–463.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Nicolas Posocco and Antoine Bonnefoy. 2021. Estimating expected calibration errors. In *Artificial Neural Networks and Machine Learning – ICANN 2021*, pages 139–150, Cham. Springer International Publishing.

Brian Thompson and Matt Post. 2020a. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020b. Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity. In *Proceedings of the Fifth Conference on Machine Translation*, pages 561–570, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Pius von Däniken, Jan Deriu, Don Tuggener, and Mark Cieliebak. 2022. On the effectiveness of automated metrics for text generation systems. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1503–1522, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Pius von Däniken, Jan Deriu, Don Tuggener, and Mark Cieliebak. 2024. Favi-score: A measure for favoritism in automated preference ratings for generative AI evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4437–4454, Bangkok, Thailand. Association for Computational Linguistics.

Johnny Wei and Robin Jia. 2021. The statistical advantage of automatic NLG metrics at the system level. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6840–6854, Online. Association for Computational Linguistics.

Siqi Wu and Paul Resnick. 2024. Calibrate-extrapolate: Rethinking prevalence estimation with black box classifiers. *Proceedings of the International AAAI Conference on Web and Social Media*, 18(1):1634–1647.

Yangjian Wu and Gang Hu. 2023. Exploring prompt engineering with GPT language models for document-level machine translation: Insights and findings. In *Proceedings of the Eighth Conference on Machine Translation*, pages 166–169, Singapore. Association for Computational Linguistics.

A Full Derivation

In Equation 4, we give the full derivation of Equation 1 in Section 2. In the following $p_k(h)$ is the density of human ratings for system π_k, $p_k(m)$ is its density of metric ratings, and $p_k(h, m)$ the joint density.

$$
\begin{aligned}
\mathbb{E}[h_k] &= \int_{-\infty}^{\infty} h p_k(h) \mathrm{d}h \\
&= \int_{-\infty}^{\infty} h \left[\int_{-\infty}^{\infty} p_k(h, m) \mathrm{d}m \right] \mathrm{d}h \\
&= \int_{-\infty}^{\infty} h \left[\int_{-\infty}^{\infty} p_k(h|m) p_k(m) \mathrm{d}m \right] \mathrm{d}h \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h p_k(h|m) p_k(m) \mathrm{d}m \mathrm{d}h \\
&= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} h p_k(h|m) \mathrm{d}h \right] p_k(m) \mathrm{d}m \\
&= \int_{-\infty}^{\infty} \mathbb{E}[h_k|m] p_k(m) \mathrm{d}m \\
&= \mathbb{E}_{p_k(m)}[\mathbb{E}_{p_k(h)}[h|m]]
\end{aligned} \tag{4}
$$

B Additional Results

Here we extend our experiment from Section 3 to additional language pairs and metrics of WMT 23. For the *en-de* language pair $N^H = 460$ and $N^M = 557$ and for *he-en* $N^H = 820$ and $H^M = 1910$. We show the results for *XCOMET* for each language pair in Tables 2, 3, and 4 (Note that Table 4 is the same as Table 1 in Section 3). We also include results for *GEMBA-MQM* (Kocmi and Federmann, 2023), which is a reference free metric based on prompting LLMs. The results can be seen in Tables 5, 6, and 7.

C Evaluating the System Dependence of WMT23 Metrics

In Section 2, we introduced the *SysDep* score. It measures the worst case in the difference of expected deviations (ED), which measures the difference between the average human rating we expect to see based on metric ratings and assuming a single global f_G and the true average human rating for a system π_k. To measure the system dependence of a metric across a set of systems $\pi_1, \ldots, \pi_K$, we compute the range of the individual ED: $SysDep = max_{\pi_k} ED(k) - min_{\pi_k} ED(k)$. We noted in Section 3 that $ED(k)$ alone is not enough to know whether system π_k will be ranked incorrectly, it depends on the true margin to the other systems, and their dependencies. By measuring the range, we consider the worst case when comparing two systems. We show the dependency ranges for all WMT23 metrics on all language pairs in Table 8. We notice that the values for *en-de* are large than the others. This is due to a larger range of human rating averages for this language pair (see also Tables 2–7 in Appendix B). We therefore also do not aggregate across language pairs.

Variants of *MetricX-23* (Juraska et al., 2023) perform best on *en-de* and *he-en*, while *GEMBA-MQM* has the lowest range for *zh-en*. The reference-free *prismSrc* (Thompson and Post, 2020a,b) metric performs worst on *en-de* and *zh-en*. The baseline *Random-sysname* (Freitag et al., 2023) performs worst for *he-en*. This baseline is an interesting case, as it is the prototypical example of a metric where every f_k is different. It assigns a fixed score to each system based on its name and adds Gaussian noise to this value to assign segment level scores. Therefore each f_k will be a different constant function.

D Intra-System Variability

In order to confirm that the observed *SysDep* scores are indeed due to a metric systematically treating systems differently and not due to variance in ratings, we will measure the maximum intra-system scores. For this, we use ratings from a single system and split them into 2 equal sized parts 10 times with different random seeds. This simulates a setting with 20 systems with half the sample size of the original setting. We then compute the *SysDep* score.

In Table 9, we show the maximum intra-system *SysDep* score computed this way over all systems for a given metric and language pair. We observe that for *he-en* and *zh-en* all scores are lower than the minimum between system *SysDep* reported in Appendix C. This confirms that in those cases metrics treat different systems differently. For the *en-de* language pair, we observe that while in many cases the intra-system score is lower than the *SysDep* between systems for the same metric and language pair, this is not always the case. This could be due to the metrics treating systems more equally for this language pair, or the relatively small sample sizes for *en-de* compared to the other language pairs.

E Estimating Conditional Expectations

In Section 3, we gave a brief overview of how to compute estimates for the functions f_k and f_G. In Listings 1 and 2, we show our *python* implementation. To estimate the system-level $\hat{f}_k$, we call the *.fit* method with human and metric ratings for system π_k. To evaluate the function $\hat{f}_k$, we use the *.conditional_expectation* method. To estimate the global function $\hat{f}_G$, we use the *.fit* method with paired human and metric ratings for all systems. We compute the remapped rating $\hat{\mu}_k^G$ by first fitting $\hat{f}_G$ and then using the *.remapped_expectation*

	Human		Metric		Remapped		Exp. Deviation
	$\hat{\mu}_k^H$	R	$\hat{\mu}_k^M$	R	$\hat{\mu}_k^G$	R	ED
GPT4-5shot	-3.724	1	0.882	2	-4.768	1	-1.044
ONLINE-W	-3.950	2	0.883	1	-4.821	2	-0.871
ONLINE-B	-4.711	3	0.871	3	-5.272	3	-0.560
ONLINE-Y	-5.643	4	0.858	4	-5.909	4	-0.266
ONLINE-A	-5.668	5	0.853	5	-6.152	5	-0.483
ONLINE-G	-6.574	6	0.834	6	-7.079	6	-0.505
ONLINE-M	-6.936	7	0.830	7	-7.399	7	-0.462
Lan-BridgeMT	-8.670	8	0.801	9	-8.670	9	-0.000
ZengHuiMT	-9.255	9	0.790	11	-9.387	11	-0.132
NLLB-Greedy	-9.543	10	0.812	8	-8.405	8	1.138
NLLB-MBR-BLEU	-10.794	11	0.797	10	-9.005	10	1.789
AIRC	-14.228	12	0.724	12	-13.658	12	0.570

Table 2: System rankings and average rating of WMT 23 *en-de* systems according to *XCOMET*.

	Human		Metric		Remapped		Exp. Deviation
	$\hat{\mu}_k^H$	R	$\hat{\mu}_k^M$	R	$\hat{\mu}_k^G$	R	ED
GPT4-5shot	-1.333	1	0.913	2	-1.690	2	-0.358
ONLINE-A	-1.381	2	0.908	3	-1.817	3	-0.436
ONLINE-B	-1.546	3	0.916	1	-1.635	1	-0.089
GTCOM-Peter	-1.886	4	0.904	4	-1.916	4	-0.030
UvA-LTL	-1.919	5	0.893	6	-2.193	6	-0.274
ONLINE-G	-2.055	6	0.895	5	-2.137	5	-0.082
ONLINE-Y	-2.349	7	0.881	8	-2.511	8	-0.162
ZengHuiMT	-2.382	8	0.889	7	-2.294	7	0.088
Samsung-Res.-Ph.	-3.234	9	0.874	9	-2.666	9	0.568
NLLB-MBR-BLEU	-3.678	10	0.869	11	-2.805	11	0.872
NLLB-Greedy	-3.790	11	0.872	10	-2.714	10	1.076
Lan-BridgeMT	-3.793	12	0.867	12	-2.823	12	0.971

Table 3: System rankings and average rating of WMT 23 *he-en* systems according to *XCOMET*.

method on the metric ratings for system π_k. We rely on the *Isotonic Regression* implementation from *scikit-learn* (Pedregosa et al., 2011) [2] and numerical utility functions from *numpy* (Harris et al., 2020).

[2] https://scikit-learn.org/stable/modules/generated/sklearn.isotonic.IsotonicRegression.html

	Human		Metric		Remapped		Exp. Deviation
	$\hat{\mu}_k^H$	R	$\hat{\mu}_k^M$	R	$\hat{\mu}_k^G$	R	ED
Lan-BridgeMT	-2.100	1	0.889	2	-2.920	2	*-0.820*
GPT4-5shot	-2.305	2	0.893	1	-2.800	1	-0.494
Yishu	-3.231	3	0.880	4	-3.179	4	0.052
ONLINE-B	-3.385	4	0.879	5	-3.188	5	0.197
HW-TSC	-3.398	5	0.883	3	-3.080	3	0.318
ONLINE-A	-3.785	6	0.856	8	-3.812	8	-0.027
ONLINE-Y	-3.792	7	0.868	6	-3.479	6	0.313
ONLINE-G	-3.857	8	0.864	7	-3.607	7	0.250
ONLINE-W	-4.062	9	0.848	9	-4.165	10	-0.103
ZengHuiMT	-4.232	10	0.846	10	-4.140	9	0.092
IOL-Research	-4.586	11	0.843	11	-4.251	11	0.335
ONLINE-M	-5.433	12	0.820	15	-4.907	15	0.526
ANVITA	-6.078	13	0.830	13	-4.602	13	1.475
NLLB-MBR-BLEU	-6.360	14	0.825	14	-4.726	14	1.634
NLLB-Greedy	-6.574	15	0.831	12	-4.578	12	**1.996**

Table 4: System rankings and average rating of WMT 23 *zh-en* systems according to *XCOMET*.

	Human		Metric		Remapped		Exp. Deviation
	$\hat{\mu}_k^H$	R	$\hat{\mu}_k^M$	R	$\hat{\mu}_k^G$	R	ED
GPT4-5shot	-3.724	1	-2.447	1	-4.123	1	-0.399
ONLINE-W	-3.950	2	-3.429	2	-4.822	2	-0.872
ONLINE-B	-4.711	3	-4.048	3	-5.383	3	-0.672
ONLINE-Y	-5.643	4	-4.424	4	-5.832	5	-0.189
ONLINE-A	-5.668	5	-4.567	5	-5.826	4	-0.158
ONLINE-G	-6.574	6	-6.018	6	-7.047	6	-0.473
ONLINE-M	-6.936	7	-6.217	7	-7.113	7	-0.177
Lan-BridgeMT	-8.670	8	-8.197	8	-8.891	9	-0.221
ZengHuiMT	-9.255	9	-8.357	9	-8.867	8	0.388
NLLB-Greedy	-9.543	10	-10.043	10	-9.683	10	-0.140
NLLB-MBR-BLEU	-10.794	11	-10.724	11	-10.352	11	0.442
AIRC	-14.228	12	-13.941	12	-12.526	12	1.702

Table 5: System rankings and average rating of WMT 23 *en-de* systems according to *GEMBA-MQM*.

	Human		Metric		Remapped		Exp. Deviation
	$\hat{\mu}_k^H$	R	$\hat{\mu}_k^M$	R	$\hat{\mu}_k^G$	R	ED
GPT4-5shot	-1.333	1	-1.923	1	-1.377	1	-0.045
ONLINE-A	-1.381	2	-3.850	2	-1.882	2	-0.501
ONLINE-B	-1.546	3	-4.108	3	-1.969	3	-0.423
GTCOM-Peter	-1.886	4	-4.859	4	-2.144	4	-0.258
UvA-LTL	-1.919	5	-5.628	6	-2.312	6	-0.393
ONLINE-G	-2.055	6	-5.240	5	-2.281	5	-0.225
ONLINE-Y	-2.349	7	-6.885	8	-2.677	8	-0.328
ZengHuiMT	-2.382	8	-6.032	7	-2.484	7	-0.102
Samsung-Res.-Ph.	-3.234	9	-8.545	12	-2.954	12	0.280
NLLB-MBR-BLEU	-3.678	10	-8.075	9	-2.817	10	0.861
NLLB-Greedy	-3.790	11	-8.261	10	-2.813	9	0.977
Lan-BridgeMT	-3.793	12	-8.469	11	-2.840	11	0.953

Table 6: System rankings and average rating of WMT 23 *he-en* systems according to *GEMBA-MQM*.

	Human		Metric		Remapped		Exp. Deviation
	$\hat{\mu}_k^H$	R	$\hat{\mu}_k^M$	R	$\hat{\mu}_k^G$	R	ED
Lan-BridgeMT	-2.100	1	-1.949	2	-2.419	2	-0.319
GPT4-5shot	-2.305	2	-1.601	1	-2.199	1	0.106
Yishu	-3.231	3	-4.790	5	-3.492	5	-0.261
ONLINE-B	-3.385	4	-4.717	4	-3.489	4	-0.104
HW-TSC	-3.398	5	-4.367	3	-3.336	3	0.062
ONLINE-A	-3.785	6	-5.568	8	-3.838	9	-0.053
ONLINE-Y	-3.792	7	-5.453	7	-3.611	6	0.181
ONLINE-G	-3.857	8	-5.275	6	-3.724	7	0.134
ONLINE-W	-4.062	9	-5.760	9	-3.772	8	0.290
ZengHuiMT	-4.232	10	-6.337	10	-4.089	11	0.143
IOL-Research	-4.586	11	-6.511	11	-4.067	10	0.519
ONLINE-M	-5.433	12	-9.115	12	-4.899	13	0.534
ANVITA	-6.078	13	-9.440	13	-4.844	12	1.234
NLLB-MBR-BLEU	-6.360	14	-11.339	15	-5.379	15	0.981
NLLB-Greedy	-6.574	15	-11.282	14	-5.312	14	1.262

Table 7: System rankings and average rating of WMT 23 *zh-en* systems according to *GEMBA-MQM*.

```python

from typing import Self, Tuple

import numpy as np
from numpy.random import Generator, default_rng

from sklearn.isotonic import IsotonicRegression


class BootstrapIsotonic:

    def __init__(
        self,
        n_bootstrap: int = 200,
        rng: int | Generator = 0xdeadbeef,
    ):
        self.n_bootstrap = n_bootstrap
        self.rng = default_rng(rng)
        self.models = []

    def fit(
        self,
        human_ratings: np.ndarray[float],  # 1d array of human ratings
        metric_ratings: np.ndarray[float], # 1d array of matching metric ratings
    ) -> Self:
        # fit a model of f_k or f_G, i.e. the conditional expectation of human ratings given metric ratings
        # to get model a system-level function f_k, use only human_ratings for that given system k
        # to model the global function f_G, use data from all systems

        assert len(human_ratings) == len(metric_ratings)
        n_samples = len(human_ratings)

        for _ in range(self.n_bootstrap):
            bootstrap_indices = self.rng.choice(
                np.arange(n_samples),
                n_samples,
                replace=True
            )
            isotonic_model = IsotonicRegression(
                y_min=None,  # MQM scores range from large negative to 0
                y_max=0.,
                increasing=True,  # metric has positive correlation
                out_of_bounds='nan',  # don't extrapolate
            )
            isotonic_model.fit(
                X=metric_ratings[bootstrap_indices],
                y=human_ratings[bootstrap_indices],
            )
            self.models.append(isotonic_model)

        return self

    def _predict_bootstrap(
        self,
        m: np.ndarray[float]  # 1d array of metric ratings
    ) -> np.ndarray[float]:
        # helper function getting predictions from each model, returns 2d array of size [n_bootstrap, len(m)]
        result = np.zeros((self.n_bootstrap, len(m)), dtype=float)
        for bix, model in enumerate(self.models):
            result[bix, :] = model.predict(m)
        return result
```

Listing 1: Part 1 of the *python* code to estimate $\hat{f}_k$, $\hat{f}_G$, and $\hat{\mu}_k^G$.

```python

    def conditional_expectation(
        self,
        metric_ratings: np.ndarray[float],  # 1d array of metric ratings
    ) -> np.ndarray[float]:
        # this computes the function f_k or f_G (depending on what data we fitted on)
        bootstrap_predictions = self._predict_bootstrap(metric_ratings)
        return np.nanmean(bootstrap_predictions, axis=0)

    def confidence(
        self,
        metric_ratings: np.ndarray[float],  # 1d array of metric ratings
    ) -> Tuple[np.ndarray[float], np.ndarray[float]]:
        # this computes the confidence bounds around f_k or f_G in Figure 1
        bootstrap_predictions = self._predict_bootstrap(metric_ratings)
        lower = np.nanpercentile(bootstrap_predictions, 2.5, axis=0)
        upper = np.nanpercentile(bootstrap_predictions, 97.5, axis=0)
        return lower, upper

    def remapped_expectation(
        self,
        metric_ratings: np.ndarray[float], # 1d array of metric ratings
    ) -> float:
        # used to compute remapped system scores in Table 1.
        expected_human_ratings = self.conditional_expectation(metric_ratings)
        return np.nanmean(expected_human_ratings)
```

Listing 2: Part 2 of the *python* code to estimate $\hat{f}_k$, $\hat{f}_G$, and $\hat{\mu}_k^G$.

	en-de	he-en	zh-en
BERTscore	7.18	1.73	3.87
BLEU	9.02	2.06	4.23
BLEURT-20	3.68	1.66	3.35
Calibri-COMET22-QE	3.68	2.06	3.30
Calibri-COMET22	4.24	1.64	3.40
chrF	8.11	1.92	4.29
COMET	4.29	1.64	3.35
CometKiwi	3.84	1.95	3.02
CometKiwi-XL	3.77	1.98	3.01
CometKiwi-XXL	3.68	1.82	2.92
cometoid22-wmt21	5.44	2.11	3.30
cometoid22-wmt22	5.17	2.09	3.21
cometoid22-wmt23	4.66	1.81	3.20
docWMT22CometDA	3.87	1.65	3.37
docWMT22CometKiwiDA	4.53	1.83	2.76
eBLEU	9.49	2.08	4.29
embed-llama	7.07	2.13	4.21
f200spBLEU	8.42	2.01	4.23
GEMBA-MQM	2.57	1.48	**1.58**
instructscore	3.59	1.53	3.68
KG-BERTScore	4.24	1.88	3.04
MaTESe	5.98	1.49	3.16
mbr-metricx-qe	3.69	1.58	2.39
MEE4	8.48	1.88	4.21
MetricX-23-b	2.26	1.29	2.81
MetricX-23-c	3.56	1.69	2.36
MetricX-23-QE-b	**2.11**	1.55	2.62
MetricX-23-QE-c	2.82	**1.21**	1.65
MetricX-23-QE	2.93	1.77	3.12
MetricX-23	2.57	1.33	3.04
mre-score-labse-regular	9.70	1.65	3.94
MS-COMET-QE-22	5.87	2.22	3.37
prismRef	8.71	1.79	3.97
prismSrc	*11.24*	2.48	*4.61*
Random-sysname	9.97	*2.52*	4.53
sescoreX	3.59	1.52	3.47
tokengram-F	8.17	1.93	4.29
XCOMET-Ensemble	2.83	1.51	2.82
XCOMET-QE-Ensemble	2.95	1.78	2.95
XCOMET-XL	3.39	1.59	3.20
XCOMET-XXL	2.71	1.48	2.99
XLsim	7.83	2.01	4.20
YiSi-1	5.95	1.60	3.65

Table 8: $SysDep$ for each metric and language pair. We show the **minimum** and *maximum* for each language pair.

	en-de	he-en	zh-en
BERTscore	2.08	0.88	1.00
BLEU	2.20	0.95	0.99
BLEURT-20	2.68	0.89	0.87
Calibri-COMET22-QE	2.97	0.82	0.85
Calibri-COMET22	2.50	0.79	0.89
chrF	2.07	0.96	0.93
COMET	2.25	0.78	0.87
CometKiwi	3.01	0.83	0.81
CometKiwi-XL	2.90	0.87	0.80
CometKiwi-XXL	2.65	0.83	0.85
cometoid22-wmt21	2.81	0.76	0.89
cometoid22-wmt22	2.74	0.73	0.81
cometoid22-wmt23	2.58	0.79	0.86
docWMT22CometDA	2.32	0.83	0.95
docWMT22CometKiwiDA	2.61	0.88	0.95
eBLEU	2.49	0.91	0.98
embed-llama	2.18	0.92	1.14
f200spBLEU	2.10	0.94	0.95
GEMBA-MQM	2.89	0.93	0.88
instructscore	2.20	0.83	0.82
KG-BERTScore	2.90	0.85	0.82
MaTESe	2.71	0.77	0.80
mbr-metricx-qe	2.52	0.84	0.79
MEE4	2.15	0.91	0.94
MetricX-23-b	2.67	0.87	0.68
MetricX-23-c	2.89	0.90	0.77
MetricX-23-QE-b	2.67	0.80	0.71
MetricX-23-QE-c	2.25	0.80	0.81
MetricX-23-QE	2.46	0.80	0.74
MetricX-23	2.34	0.90	0.63
mre-score-labse-regular	2.27	0.91	0.93
MS-COMET-QE-22	2.17	0.93	0.90
prismRef	2.24	0.83	0.99
prismSrc	2.09	0.88	0.91
Random-sysname	2.36	0.96	0.95
sescoreX	2.20	0.86	0.86
tokengram-F	2.10	0.96	0.94
XCOMET-Ensemble	2.44	0.71	0.68
XCOMET-QE-Ensemble	2.30	0.72	0.72
XCOMET-XL	2.44	0.74	0.67
XCOMET-XXL	2.35	0.74	0.69
XLsim	2.71	0.90	0.94
YiSi-1	2.44	0.83	0.90

Table 9: Maximum intra-system *SysDep* score for all metrics and language pairs.

Call for Rigor in Reporting Quality of Instruction Tuning Data

Hyeonseok Moon, Jaehyung Seo, Heuiseok Lim[†]
[1]Department of Computer Science and Engineering, Korea University
{glee889,seojae777,limhseok}@korea.ac.kr

Abstract

Instruction tuning is crucial for adapting large language models (LLMs) to align with user intentions. Numerous studies emphasize the significance of the quality of instruction tuning (IT) data, revealing a strong correlation between IT data quality and the alignment performance of LLMs. In these studies, the quality of IT data is typically assessed by evaluating the performance of LLMs trained with that data. However, we identified a prevalent issue in such practice: hyperparameters for training models are often selected arbitrarily without adequate justification. We observed significant variations in hyperparameters applied across different studies, even when training the same model with the same data. In this study, we demonstrate the potential problems arising from this practice and emphasize the need for careful consideration in verifying data quality. Through our experiments on the quality of LIMA data and a selected set of 1,000 Alpaca data points, we demonstrate that arbitrary hyperparameter decisions can make any arbitrary conclusion.

1 Introduction

Instruction Tuning (IT) is a widely adopted strategy for enabling a human-interactive use of the knowledge embedded in large language models (LLMs) (Cao et al., 2023; Wang et al., 2024). By training with datasets composed of instruction-response pairs, LLM can attain the ability to generate appropriate responses to given instructions (Dubois et al., 2023; Zheng et al., 2023; Xu et al., 2023; Conover et al., 2023).

In implementing IT, data quality is considered a critical factor (Zhou et al., 2023a; Wang et al., 2024; Zhao et al., 2024b; Lu et al., 2024). Several studies have proven that selectively using high-quality IT data for training leads to better alignment performance than using the entire dataset (Liu et al., 2024b; Chen et al., 2024; Zhao et al., 2024a; Mekala et al., 2024).

Traditionally, the quality of IT data is measured by evaluating the performance of models trained on it (Liu et al., 2024b; Chen et al., 2024; Zhao et al., 2024a; Xia et al., 2024a). This approach stems from the consensus that *data is deemed good if it produces a good model*. Consequently, most studies on data quality establish a training configuration for models that represent data quality. Then, the performance of the trained model is regarded as the data quality. (Zhou et al., 2023a; Zhao et al., 2024a; Xia et al., 2024b; Du et al., 2023; Zhou et al., 2023b).

However, we observed that these studies often lack justification for the hyperparameter settings used in model training. Table 1 presents the diverse hyperparameter configurations utilized in previous research implementing IT with a sampled 1K general domain IT dataset. We discovered that the configurations may vary across studies, even when training the same model with identical data sizes.

In this study, we question whether reaching coherent conclusions under varying settings is possible. Specifically, we emphasize that conclusions regarding data quality can easily be altered based on arbitrarily chosen hyperparameter settings. For instance, even if one might report that dataset A is superior to dataset B, another could claim that B is better by training models under different settings, even with the same dataset, model, and test settings. This variability poses a risk of causing significant confusion.

As a representative case, we consider two general-domain IT datasets: LIMA (Zhou et al., 2023a) and sampled 1K dataset from Alpaca (Alpaca-longest (Zhao et al., 2024a)). In (Zhao et al., 2024a), it was reported that a model trained on Alpaca-longest outperformed a model trained on LIMA. However, our experiments contrarily demonstrate that LIMA can also be regarded as better than Alpaca-longest, depending on the selected training setting. Given the current research trend

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 100–109
July 27 - August 1, 2025

Paper	Epochs	LR	LR Scheduler	Batch	Data Pool
*Training **Llama-2-7B** with sampled **1K** general domain IT data*					
Ghosh et al. (2024)	3	5e-5	-	32	Lima
Raghavendra et al. (2024)	3	1e-5	-	8	Dolly
Yu et al. (2024)	3	1e-5	-	64	Alpaca / WizardLM
Du et al. (2023)	3	2e-5	Cosine	128	Alpaca+HC3+WizardLM+Dolly+Self-Instruct+Lima
Li et al. (2024)	3	2e-5	-	128	Alpaca / WizardLM
Liu et al. (2024a)	3	2e-5	Cosine	128	Alpaca-gpt4 / Lima
Mekala et al. (2024)	3	2e-5	Cosine	128	Alpaca / Dolly
Kong et al. (2024)	8	1e-5	Cosine	64	Lima
Zhao et al. (2024a)	15	1e-5	Linear	128	Alpaca / WizardLM / Lima
Zhou et al. (2023a)	15 (ES)	1e-5	Linear	64	Lima
*Training **Llama-2-13B** with 1K general domain IT data*					
Ghosh et al. (2024)	3	5e-5	-	32	Lima
Zhao et al. (2024b)	10	1e-4	-	16	Alpaca-gpt4
Liu et al. (2024a)	3	2e-5	Cosine	128	Alpaca-GPT4 / Lima
Mekala et al. (2024)	3	2e-5	Cosine	128	Alpaca / Dolly
Zhao et al. (2024a)	15	1e-5	Linear	128	Alpaca / WizardLM / Lima
*Training **Mistral 7B** with 1K general domain IT data*					
Kong et al. (2024)	4	1e-5	Cosine	64	Lima
Zhao et al. (2024a)	15	2e-6	Linear	128	Alpaca / WizardLM / Lima
Ghosh et al. (2024)	3	5e-5	-	32	Lima
Yu et al. (2024)	3	1e-5	-	64	Alpaca / WizardLM
Yin et al. (2024)	4	4e-6	-	128	WizardLM / UltraChat / ShareGPT

Table 1: Hyperparameters reported by previous studies, adopted to train LLMs with 1K general domain IT data. The data pool details the sources from which the 1K data samples were drawn. Detailed descriptions of these data pools are provided in the Table 4. The '+' symbol indicates experiments where samples were drawn from a combined data mix of all mentioned datasets. The '/' symbol reports studies that sampled individually from each data pool.

of arbitrarily determining hyperparameters for validation models, this confusion can be identified as a severe yet persistent problem.

Through our experiments, we emphasize the necessity of rigor in reporting data quality. Furthermore, our discussion suggests the importance of identifying (at least) locally optimal hyperparameters and reporting data quality under these settings.

2 Related Works

Previous research has widely acknowledged the importance of data quality in performing IT. Chen et al. (2024) proposed that training LLMs with a small, carefully selected subset of high-quality data can significantly improve alignment performance within the vast IT data pool. Furthermore, Zhou et al. (2023a) even suggested that carefully curated high-quality 1,000 data points are sufficient to attain alignment performance for LLMs. Motivated by these findings, numerous studies are exploring various methodologies focused on selecting high-quality instruction tuning data (Wang et al., 2024; Chen et al., 2024; Zhao et al., 2024a; Xia et al., 2024b; Lu et al., 2024; Liu et al., 2024b).

However, most studies lack justification for the selected hyperparameter setting to train verification models. Consequently, the training setups become diversified even when using the same LLM and dataset. We argue that the importance of selecting appropriate hyperparameters has long been emphasized (Yu and Zhu, 2020; McCandlish et al., 2018; Halfon et al., 2024). The community widely recognizes that optimal hyperparameters are often specific to particular LLMs and datasets, and reported performance may vary based on the experimental setup (Van Rijn and Hutter, 2018; Jin, 2022; Gkouti et al., 2024; Bi et al., 2024). However, we find that research on data quality frequently reports performance without adequately considering these factors.

In this study, we highlight the potential confusion that can result from neglecting these considerations and demonstrate the necessity of a rigorous experimental setup to report data quality.

3 Experimental Setting

3.1 Exam-taker Dataset

In our experiments, we adopt two general domain IT datasets, each comprising 1,000 samples, as our exam-taker datasets. By comparing the quality of these two datasets, we examine how the judgment on the exam-taker datasets varies with different arbitrarily chosen hyperparameter settings.

LIMA (Zhou et al., 2023a) LIMA is a high-quality dataset comprising 1,000 IT data points,

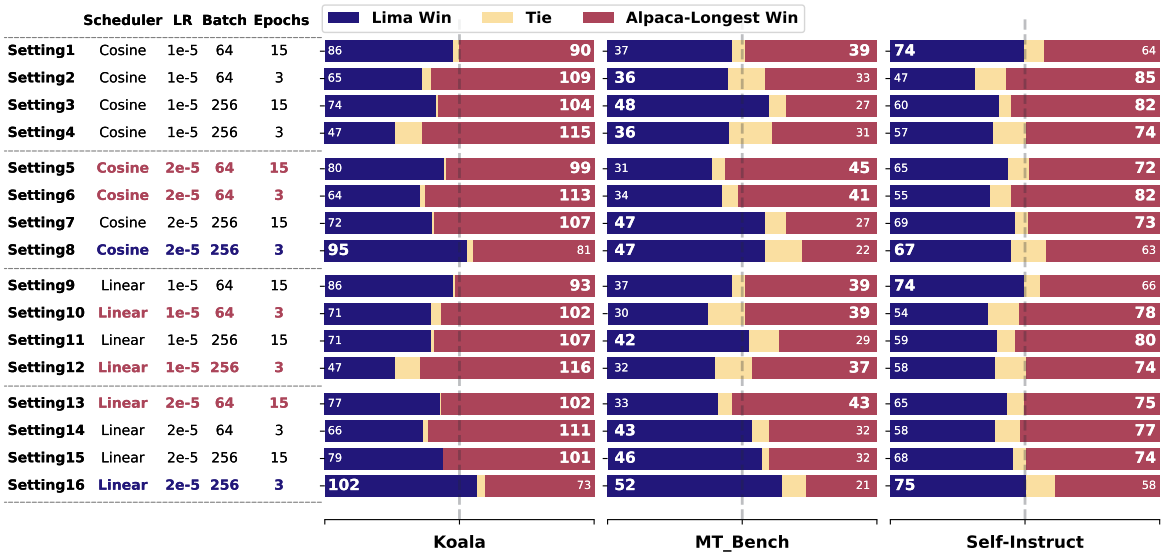


Figure 1: The performance comparison between the two models trained with LIMA and Alpaca-Longest. We train Llama-2-7B model with each dataset, We evaluate the data quality when training each dataset with the Llama-2-7B model. is depicted on the Y-axis represents the hyperparameter settings used in each experiment. We bolded the settings that consistently demonstrated conclusive results across all three evaluation datasets.

carefully curated by human efforts with an emphasis on quality and diversity.

Alpaca-Longest (Zhao et al., 2024a) Zhao et al. (2024a) selected the 1,000 entries with the longest token lengths from the Alpaca dataset (Taori et al., 2023). This approach proved more effective than training on the entire Alpaca dataset and significantly outperformed other baselines such as Alpagasus (Chen et al., 2024). According to the original paper, training with this data resulted in higher alignment performance than LIMA.

3.2 Experimental Model

The quality of the Exam-taker Dataset is determined by the performance of the experimental model trained on it. We conduct experiments using the Llama-2-7B model (Touvron et al., 2023) and the Mistral-7B-v0.3 model (Jiang et al., 2023). The main paper reports the results for the Llama-2-7B model, and Appendix A includes the results for the Mistral-7B model.

3.3 Experimental Setting

This study focuses on four commonly reported hyperparameters: learning rate, learning rate scheduler, batch size, and number of epochs. We report the experimental results obtained from varying these parameters. We conduct comparative experiments for each setting by choosing two prevalent yet distinct values. While numerous other potential variations exist, such as weight decay and dropout, we leave these for future exploration. Apart from the hyperparameters under investigation, detailed experimental settings are provided in Appendix C.

3.4 Test Dataset

To evaluate the performance of the trained model, we use three LLM alignment benchmarks: Koala (Geng et al., 2023), MT-Bench (Zheng et al., 2023), and Self-Instruct (Wang et al., 2023). These benchmarks serve as an instruction-following evaluation tool, assessing LLMs by evaluating the quality of text generated in response to given instructions. We employ GPT-4o (Hurst et al., 2024)[1] as a judge to compare the performance of experimental models for each benchmark. The judge prompts used in the experiments are detailed in Appendix D.

4 Experimental Results

4.1 LIMA vs Alpaca-Longest

Figure 1 presents the experimental results based on hyperparameter variations. Our results show that if we choose specific settings (*e.g.,* Settings 4, 5, 10, 12, 13), we can report that Alpaca-longest exhibits superior data quality compared to LIMA. At the same time, if we choose other configurations

[1]https://openai.com/index/hello-gpt-4o/

Dataset	LIMA			Alpaca-Longest		
Comparison with Setting 1	Setting 1 Wins	Tie	Setting x Wins	Setting 1 Wins	Tie	Setting x Wins
vs **Setting 2**	**112**	10	58	**121**	7	52
vs **Setting 3**	**101**	7	72	**97**	10	73
vs **Setting 4**	**142**	7	31	**144**	10	26
vs **Setting 5**	**98**	9	73	**102**	6	72
vs **Setting 6**	**114**	6	60	**109**	7	64
vs **Setting 7**	83	8	**89**	73	7	**100**
vs **Setting 8**	**121**	8	51	**136**	9	35
vs **Setting 9**	**86**	10	84	**84**	14	82
vs **Setting 10**	**116**	9	55	**130**	9	41
vs **Setting 11**	**101**	5	74	**96**	9	75
vs **Setting 12**	**146**	7	27	**145**	9	26
vs **Setting 13**	**96**	7	77	**102**	8	70
vs **Setting 14**	**124**	8	48	**114**	5	61
vs **Setting 15**	82	7	**91**	78	6	**96**
vs **Setting 16**	**109**	10	61	**145**	5	30

Table 2: We report the performance of the Llama-2-7B model, trained under each setting, as evaluated on the Koala dataset. Details for each setting are presented in the Figure 1.

(*e.g.,* Settings 8, 16), we can report that LIMA still demonstrates higher data quality.

Considering that authors have determined such hyperparameters arbitrarily, this represents a significant concern. We view that the ability to alter reported conclusions based on subjective decisions can severely undermine the reliability of scientific discussions.

4.2 Among the Same Dataset

Then, which setting should we choose to report? Considering that the primary goal of the IT dataset is to construct high-performance models, it would be reasonable and practical to report results based on the best achievable performance with the given data (Koehn et al., 2018, 2020; Budach et al., 2022; Van Rijn and Hutter, 2018).

In this section, we identify the optimal settings among the configurations tested. We recognize that other configurations with better performance may have been overlooked. We focus on local optimality within our considered settings and discuss its implications. Figure 2 compares model performance across various hyperparameter settings, using **Setting1** as the baseline.

Our experiments reveal that **Setting7, 15** (2e-5 LR / 256 Batch / 15 Epochs) maximizes model performance within our study. Notably, we can find that such configurations are far beyond the widely chosen settings in existing research. As our brief survey in Table 1 indicates, most studies opt to train Llama-2-7B for only three epochs when using 1K IT datasets. However, our results show that this setup yielded significantly lower performance than training for 15 epochs under the same conditions. This finding suggests that the reported performance in many studies may reflect the under-trained performance of models, which may fail to fully represent the potential of the exam-taker dataset.

5 Discussion

We argue that it is inevitable to evaluate the downstream model performance. We acknowledge that assessing data quality through the performance of a trained model can be ambiguous. However, we also argue that the quality of training data must inevitably be assessed through the model's performance after training.

We would like to discuss how the quality of training data is generally acknowledged. The goal of constructing the training dataset is to develop a model that aligns with the intended purpose. Thus, in terms of training data, "good data" is defined as data that produces a "good model"(Chen et al., 2024; Koehn et al., 2020). This fundamentally differs from constructing benchmark datasets for evaluation. Since training data's primary aim is to build a strong model, data that appear high-quality to humans (or any frontier LLMs) may offer little value if the trained model's performance remains subpar (Liu et al., 2024b).

In this context, the quality of training data is fundamentally linked to the performance of the model trained on it. While there are various plausible methods to assess training data, these methods might remain indirect indicators, without validating with the performance of the trained model.

Consequently, most studies demonstrate the quality of the data under evaluation by training it onto one (possibly several) model and reporting their performance. We do not consider this approach erroneous; instead, we view it as a natural and inevitable choice. Our stance is that if model-based verification is unavoidable, a more thorough and rigorous training configuration would be essential to verify data quality. Our experiments demonstrate that hyperparameters can introduce unintended biases that skew the objective evaluation of the data quality.

We argue that authors researching data quality have responsibility for such validation. To address these ambiguities, we suggest selecting the hyperparameter setting that yields the highest performance within a given data and reporting the model's performance under this setting. This ap-

proach seeks to evaluate the model by maximizing the data's potential.

Given that hyperparameter search is being performed in relatively small-sized PLMs (e.g., BERT (Devlin et al., 2019), BART (Lewis et al., 2020)) (Latif and Kim, 2024; Ljubešić et al., 2024; Roele, 2021), we argue that it is challenging to justify its omission in LLMs other than its high cost. Even when researchers do not conduct their own hyperparameter search, there have been multiple attempts to use existing configurations (Zhou et al., 2023b). However, as shown in Table 1 with the example of Mistral, there appears to be no established standard configuration when tuning relatively recent LLMs.

Reporting the best performance would certainly require additional costs for experiments, but we believe this is a necessary sacrifice to strengthen scientific discourse. We argue that arbitrary conclusions stemming from arbitrary hyperparameter choices pose a greater risk than incurring additional costs. While a comprehensive hyperparameter search may not always be necessary, we claim that authors should clearly justify their chosen hyperparameters. Even if they do not report peak performance, employing the best settings from our paper or established training configurations (ex. LIMA configuration) would still be a rational approach.

6 Conclusion

In our examination of various studies addressing data quality, we observed a recurring issue where researchers often arbitrarily select hyperparameters when training models to verify data quality. Our experiments reveal that arbitrary hyperparameter choices can lead to arbitrary conclusions. Moreover, we found that hyperparameters chosen without justification often fail to achieve optimal performance on the exam-taker datasets, resulting in unreliable conclusions. To address this, we propose establishing a local hyperparameter pool and training models under locally optimal settings within this pool. While additional costs for hyperparameter validation are inevitable, we consider this a necessary sacrifice for attaining the reliability of scientific discourse. To ensure rigorous reporting and sustainable consensus, we urge careful attention.

Limitation

The numerous hyperparameter settings we did not consider may remain a limitation of our study. Within our budget constraints, we verified as many possibilities as possible. Fortunately, we found significant variations in model performance even within the four factors we examined, allowing us to draw generalized conclusions. Through our brief survey, we can also found that various other hyperparameter variants, such as weight decay, warmup steps, are also introduced without justification. Exploring additional possibilities to identify an optimal setup could be a meaningful area for future research.

In this study, we did not attempt hyperparameter optimization (HPO), as finding optimal values was outside the scope of our research. Applying HPO when reporting on data quality, could serve as an excellent direction for future research.

Although we conducted experiments using only two datasets, we do not see this as a limitation. We believe this setup clearly demonstrates the inherent ambiguity in reporting data quality. Possibly numerous other datasets can exist where hyperparameter settings could alter reporting conclusions. Instead of identifying additional dataset pairs, we consider it more valuable to focus future research on strategies to mitigate such ambiguities.

Ethics Statement

We do not challenge the quality assessment results reported by the Zhao et al. (2024a), which proposed Alpaca-Longest and estimated that Alpaca-Longest is better than LIMA. Based on the results from our experiments using locally optimal settings within our pool, Alpaca-Longest can arguably be considered superior to Lima. This finding aligns with the results reported in the original paper. In all experiments, we exclusively used artifacts approved for research purposes.

Acknowledges

This work was partly supported by ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (IITP-2025-RS-2020-II201819, 25%), Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (RS-2024-00398115, Research on the reliability and coherence of outcomes produced by Generative AI, 25%), Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Min-

istry of Education(NRF-2021R1A6A1A03045425, 25%), and Culture, Sports and Tourism R&D Program through the korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Toruism in 2024 (Project Name : Development of generative AI-based publishing content analysis and sharing platform technology to respond to changes in the publishing environment. Project Number : : RS-2024-00442061, 25%)

References

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. Deepseek llm: Scaling open-source language models with longtermism. *CoRR*.

Lukas Budach, Moritz Feuerpfeil, Nina Ihde, Andrea Nathansen, Nele Noack, Hendrik Patzlaff, Felix Naumann, and Hazar Harmouch. 2022. The effects of data quality on machine learning performance. *arXiv preprint arXiv:2207.14529*.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. 2023. Instruction mining: Instruction data selection for tuning large language models. *arXiv preprint arXiv:2307.06290*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. *Company Blog of Databricks*.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore. Association for Computational Linguistics.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. 2023. Mods: Model-oriented data selection for instruction tuning. *arXiv preprint arXiv:2311.15653*.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. *Blog post, April*, 1:6.

Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Ramaneswaran S, Deepali Aneja, Zeyu Jin, Ramani Duraiswami, and Dinesh Manocha. 2024. A closer look at the limitations of instruction tuning. In *Forty-first International Conference on Machine Learning*.

Nefeli Gkouti, Prodromos Malakasiotis, Stavros Toumpis, and Ion Androutsopoulos. 2024. Should I try multiple optimizers when fine-tuning a pre-trained transformer for NLP tasks? should I tune their hyperparameters? In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2555–2574, St. Julian's, Malta. Association for Computational Linguistics.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Alon Halfon, Shai Gretz, Ofir Arviv, Artem Spector, Orith Toledo-Ronen, Yoav Katz, Liat Ein-Dor, Michal Shmueli-Scheuer, and Noam Slonim. 2024. Stay tuned: An empirical study of the impact of hyperparameters on llm tuning in real-world applications. *arXiv preprint arXiv:2407.18990*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Honghe Jin. 2022. Hyperparameter importance for machine learning algorithms. *arXiv preprint arXiv:2201.05132*.

Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, et al. 2019. A study of bfloat16 for deep learning training. *arXiv preprint arXiv:1905.12322*.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the wmt 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 739–752, Belgium, Brussels. Association for Computational Linguistics.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Jiaming Zhou, and Haoqin Sun. 2024. Self-prompt tuning: Enable autonomous role-playing in llms. *CoRR*, abs/2407.08995.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Atif Latif and Jihie Kim. 2024. Evaluation and analysis of large language models for clinical text augmentation and generation. *IEEE Access*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7602–7635, Mexico City, Mexico. Association for Computational Linguistics.

Liangxin Liu, Xuebo Liu, Derek F Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. 2024a. Selectit: Selective instruction tuning for large language models via uncertainty-aware self-reflection. *arXiv preprint arXiv:2402.16705*.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024b. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Nikola Ljubešić, Vít Suchomel, Peter Rupnik, Taja Kuzman, and Rik van Noord. 2024. Language models on a diet: Cost-efficient development of encoders for closely-related languages via additional pretraining. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages @ LREC-COLING 2024*, pages 189–203, Torino, Italia. ELRA and ICCL.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2024. #instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *The Twelfth International Conference on Learning Representations*.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. 2018. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*.

Dheeraj Mekala, Alex Nguyen, and Jingbo Shang. 2024. Smaller language models are capable of selecting instruction-tuning training data for larger language models. *arXiv preprint arXiv:2402.10430*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Mohit Raghavendra, Vaskar Nath, and Sean Hendryx. 2024. Revisiting the superficial alignment hypothesis. *arXiv preprint arXiv:2410.03717*.

Cees Roele. 2021. WVOQ at SemEval-2021 task 6: BART for span detection and classification. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 270–274, Online. Association for Computational Linguistics.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Dusan Stosic and Paulius Micikevicius. 2021. Accelerating ai training with nvidia tf32 tensor cores. *NVIDIA Technical Blog*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jan N Van Rijn and Frank Hutter. 2018. Hyperparameter importance across datasets. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2367–2376.

Jiahao Wang, Bolin Zhang, Qianlong Du, Jiajun Zhang, and Dianhui Chu. 2024. A survey on data selection for llm instruction tuning. *arXiv preprint arXiv:2402.05123*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024a. LESS: Selecting influential data for targeted instruction tuning. In *Forty-first International Conference on Machine Learning*.

Tingyu Xia, Bowen Yu, Kai Dang, An Yang, Yuan Wu, Yuan Tian, Yi Chang, and Junyang Lin. 2024b. Rethinking data selection at scale: Random selection is almost all you need. *arXiv preprint arXiv:2410.09335*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Mingjia Yin, Chuhan Wu, Yufei Wang, Hao Wang, Wei Guo, Yasheng Wang, Yong Liu, Ruiming Tang, Defu Lian, and Enhong Chen. 2024. Entropy law: The story behind data compression and llm performance. *CoRR*, abs/2407.06645.

Simon Yu, Liangyu Chen, Sara Ahmadian, and Marzieh Fadaee. 2024. Diversify and conquer: Diversity-centric data selection with iterative refinement. *arXiv preprint arXiv:2409.11378*.

Tong Yu and Hong Zhu. 2020. Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*.

Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. 2024a. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning. In *Forty-first International Conference on Machine Learning*.

Yingxiu Zhao, Bowen Yu, Binyuan Hui, Haiyang Yu, Minghao Li, Fei Huang, Nevin L. Zhang, and Yongbin Li. 2024b. Tree-instruct: A preliminary study of the intrinsic relationship between complexity and alignment. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16776–16789, Torino, Italia. ELRA and ICCL.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 55006–55021. Curran Associates, Inc.

Daquan Zhou, Kai Wang, Jianyang Gu, Xiangyu Peng, Dongze Lian, Yifan Zhang, Yang You, and Jiashi Feng. 2023b. Dataset quantization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17205–17216.

A Experimental Results - Mistral

We conducted the same experiments described in Section 4 using the Mistral-7B model. The results are reported in Table 3 and Figure 2.

Dataset	LIMA			Alpaca-Longest		
Comparison with Setting 1	Setting 1 Wins	Tie	Setting x Wins	Setting 1 Wins	Tie	Setting x Wins
vs **Setting 2**	79	12	**89**	58	11	**111**
vs **Setting 3**	75	12	**93**	50	14	**116**
vs **Setting 4**	**100**	8	72	**107**	14	59
vs **Setting 5**	**116**	13	51	**126**	13	41
vs **Setting 6**	**97**	17	66	**95**	15	70
vs **Setting 7**	**93**	12	75	70	14	**96**
vs **Setting 8**	**125**	8	47	**130**	12	38
vs **Setting 9**	**87**	16	77	**82**	18	80
vs **Setting 10**	**83**	14	**83**	58	15	**107**
vs **Setting 11**	81	7	**92**	45	6	**129**
vs **Setting 12**	**102**	10	68	**96**	9	75
vs **Setting 13**	**124**	13	43	**129**	18	33
vs **Setting 14**	**96**	22	62	**87**	16	77
vs **Setting 15**	**88**	14	78	78	14	**88**
vs **Setting 16**	**109**	9	62	**118**	10	52

Table 3: We report the performance of the Mistral-7B model, trained under each setting, as evaluated on the Koala dataset. Details for each setting are presented in the Figure 1.

As shown in Table 3, models trained for 15 epochs generally outperformed those trained for only 3 epochs, even within the same settings. This finding suggests that commonly adopted hyperparameter settings in prior research may not be optimal and that reported performance might not fully exploit the data's potential.

Figure 2 illustrates the potential conclusions we can draw from various settings using Mistral. There is still significant diversity between settings, supporting our earlier conclusions in Section 4. We demonstrate that merely using multiple models is insufficient to enhance robustness in data quality validation, emphasizing the necessity of hyperparameter generalization.

B Dataset Details

Dataset	Paper / Description	Data Size
Alpaca	Taori et al. (2023)	52K
Alpaca-GPT4	Peng et al. (2023)	52K
Dolly	Conover et al. (2023)	15K
HC3	Guo et al. (2023)	24.3K
ShareGPT	Chiang et al. (2023)	52K
UltraChat	Ding et al. (2023)	200K
WizardLM	Xu et al. (2023)	700K

Table 4: We report only the data aimed at performing IT in a general domain, which are adopted to previous studies. Each dataset consists of a pair, featuring a human instruction and an appropriate response.

C Experimental Details

We conducted experiments with a weight decay of 0.0, a warmup of 0.0, and a maximum length of 2,048, utilizing the `HuggingFace` trainer (Wolf et al., 2020). To enhance learning efficiency, we applied `bf16` (Kalamkar et al., 2019) and `tf32` (Stosic and Micikevicius, 2021) strategy. All training was performed using FlashAttention-2 (Dao et al., 2022) and DeepSpeed Stage 2 (Smith et al., 2022). For inference, we employed vllm (Kwon et al., 2023). Our setup included four RTX-A6000 GPUs with 48GB each for model training and inference. The original batch size per GPU was set to 2, and we used gradient accumulation to increase the batch size. Other settings followed the default configurations provided by the `HuggingFace` trainer.

D LLM-as-a-Judge

System Prompt
Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below.
You should choose the assistant that follows the user's instructions and answers the user's question better.
Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of their responses.
Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision.
Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible.
After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.
Input Statements
You are a helpful and precise assistant for checking the quality of the answer.
[Question]
{question}
[The Start of Assistant 1's Answer]
{Response From Assistant 1}
[The End of Assistant 1's Answer]
[The Start of Assistant 2's Answer]
{Response From Assistant 2}
[The End of Assistant 2's Answer]

Table 5: Prompt used for training the LLM: For models not supporting system prompts, we combined the system prompt and user prompt into a single input statement.

The prompt we used is presented in Table 5. In

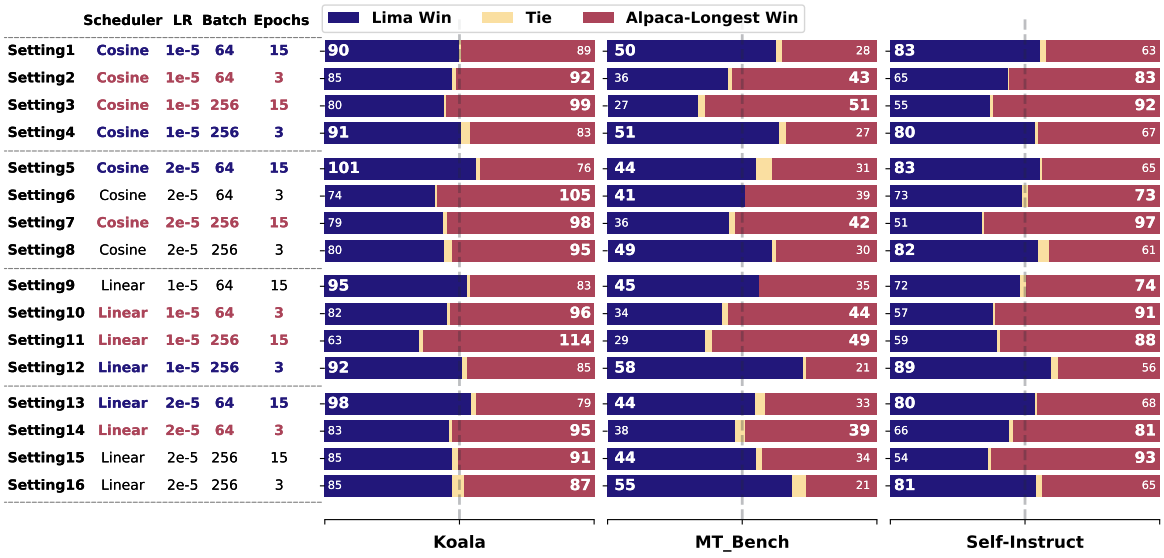


Figure 2: The performance comparison between the two models trained with LIMA and Alpaca-Longest. We train Mistral-7B model with each dataset, We evaluate the data quality when training each dataset with the Mistral-7B model. is depicted on the Y-axis represents the hyperparameter settings used in each experiment. We bolded the settings that consistently demonstrated conclusive results across all three evaluation datasets.

all our experiments, we randomize the order of presented responses to relieve any unintended effects driven by the positional bias. We conducted our experiments with GPT-4o (`gpt-4o-2024-08-06`), setting the temperature to 0 and top-p to 1.0. The API usage cost for the experiments detailed in Table 2 was $11.25. Conducting a similar hyperparameter search using GPT4o-mini incurred a cost of $1.44. While using GPT4o-mini presents a cost-effective option, the Pearson-r correlation score between GPT4o and GPT4o-mini was 0.559 in our experiments. Although this score might be considered reasonably high, we argue using GPT-4o is more effective for establishing a more precise and rigorous setting. We leave experiments with alternative judges for future research.

BQA: Body Language Question Answering Dataset for Video Large Language Models

Shintaro Ozaki **Kazuki Hayashi** **Miyu Oba**
Yusuke Sakai **Hidetaka Kamigaito** **Taro Watanabe**
Nara Institute of Science and Technology (NAIST), Japan
ozaki.shintaro.ou6@naist.ac.jp
{sakai.yusuke.sr9, kamigaito.h, taro}@is.naist.jp

Abstract

A large part of human communication relies on nonverbal cues such as facial expressions, eye contact, and body language. Unlike language or sign language, such nonverbal communication lacks formal rules, requiring complex reasoning based on commonsense understanding. Enabling current Video Large Language Models (VideoLLMs) to accurately interpret body language is a crucial challenge, as human unconscious actions can easily cause the model to misinterpret their intent. To address this, we propose a dataset, BQA, a body language question answering dataset, to validate whether the model can correctly interpret emotions from short clips of body language comprising 26 emotion labels of videos of body language. We evaluated various VideoLLMs on the BQA with and without Multimodal Chain of Thought (CoT) and revealed that understanding body language is challenging, and our analyses of the wrong answers by VideoLLMs show that certain VideoLLMs made largely biased answers depending on the age group and ethnicity of the individuals. We also found consistent error patterns in VideoLLMs[1].

1 Introduction

Video large language models (VideoLLMs) (Wang et al., 2024; Ye et al., 2025; Zhang et al., 2024a; Team et al., 2024) process videos by integrating multimodal inputs into an understanding of the content. These models take video frames, sound, and accompanying text as input and generate text, answers to questions (Maaz et al., 2024; Lei et al., 2018), or predictions based on the video (Xiao et al., 2021; Yi et al., 2020), enabling various applications, such as video summarization and question answering. This capability fosters a future where humans and models coexist, making it essential for VideoLLMs to grasp human emotions and body language for interaction. One study (Hyun et al., 2024) has investigated emotion detection from body language, identifying smiles and their underlying causes. However, since this approach is limited to analyzing a single emotion, it remains unclear whether the findings are generalized to all human emotions. If VideoLLMs are unable to understand human emotion from body language, they may not be suitable for future applications such as dialogue systems and AI robots, where emotional awareness is crucial for enabling more effective interactions.

Our research focuses on the analysis of various emotional expressions in human body language. We created a dataset called *BQA*, a multiple-choice QA task, in which each body language video is associated with a question regarding a particular emotion comprising four choice answers, e.g., Surprise, Confidence, Anger, and Embarrassment, reformatting the Body Language Dataset created for pose estimation (Luo et al., 2020). The BQA consists of 7,632 short videos (5–10 seconds, 25 fps), depicting human body language with metadata (gender, age, ethnicity) and 26 emotion labels per video. The BQA creation involves four steps using Gemini (Team et al., 2024): extracting answer choices, generating questions, evaluating potential harm, and assigning difficulty labels. Moreover, we evaluated recent VideoLLMs on BQA, with and without Chain of Thought (CoT) (Zhang et al., 2024b) as well as conducted multiple human evaluations, and found the task enough challenging for VideoLLMs. Error analysis revealed biases toward specific ages or ethnicities, and Multimodal CoT, despite improving accuracy, showed consistent error patterns.

2 Body Language Dataset (BoLD)

Body Language Dataset (BoLD) (Luo et al., 2020) is a dataset for recognizing human actions and selecting appropriate emotions created by splitting

[1]The dataset is available at https://huggingface.co/datasets/naist-nlp/BQA.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 110–123
July 27 - August 1, 2025

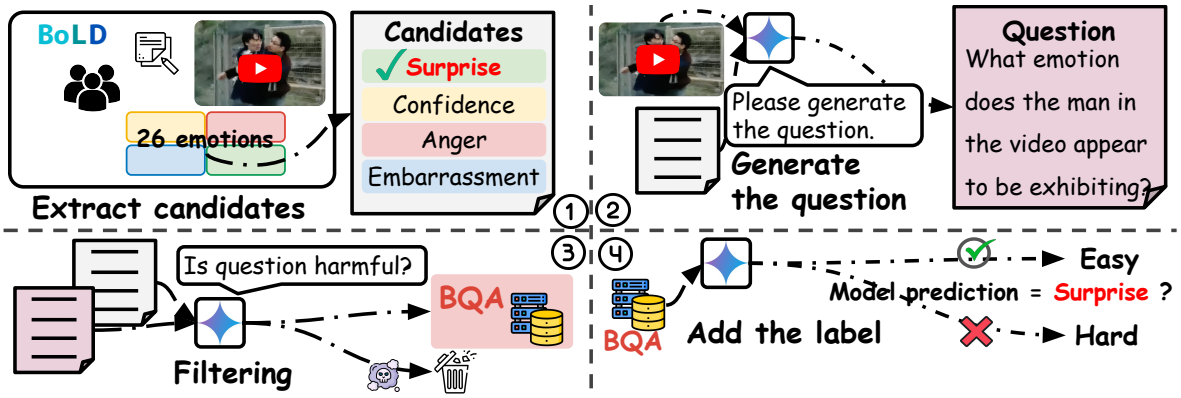


Figure 1: 4 steps of creating the BQA dataset. In STEP1, candidates are created; in STEP2, questions are generated; in STEP3, filtering is conducted; and in STEP4, we assign (Easy/Hard) labels, describing the details in Section 3.

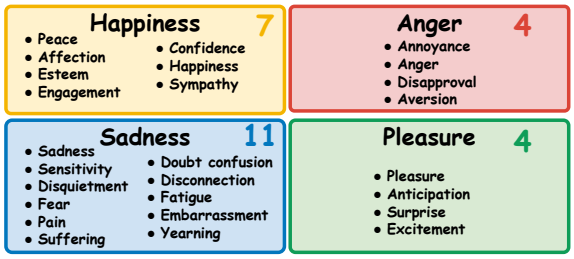


Figure 2: Categorized the 26 emotion labels into 4 groups with similar emotions to extract the candidates.

150 films, totaling 220 hours of footage, resulting in 9,876 video clips. Each clip is approximately 5 seconds long, comprising nearly 125 frames. These videos were annotated with 26 emotion labels (Kosti et al., 2017) via crowdsourcing, where multiple annotators assigned emotion labels on a 10-point scale, which were normalized to represent the emotion of each video. Metadata, including the age, gender, and ethnicity of the individuals in the clips, is also available. However, BoLD is designed for a model directly predicting the emotion, and is not suitable for prompting with clear answers in order to investigate the capacity of VideoLLMs, which expect inference with natural language.

3 Dataset Construction

We transformed BoLD into a multiple-choice QA format by generating questions from video content and using the 26 emotion labels as answers to evaluate how well VideoLLMs understand human emotions expressed through body language, adding steps to extract appropriate choices since BoLD was not designed for LLMs evaluation. To design a QA task for evaluating the LLMs' understanding of body language for emotional expression, we followed the approach of mCSQA (Sakai et al., 2024a), which semi-automatically generates QA questions based on candidate answers using LLMs. The whole process consists of four steps as described in Figure 1. First, we extract candidate choices from the BoLD's metadata (Figure 1-1) followed by question generation using a Gemini based on the video and the candidate choices (Figure 1-2). Then, we automatically filter out inappropriate QAs (Figure 1-3). Lastly, we let the Gemini solve the QAs to evaluate difficulty levels (Figure 1-4).

STEP1: Extract Candidates We categorized 26 emotion labels defined in BoLD into 4 groups: Happiness, Anger, Sadness, and Pleasure, as shown in Figure 2, based on the research in which emotions are classified into four main groups (James, 1890). For the creation of BQA, we apply a multiple-choice question format, where the one with the highest empathy level is treated as correct, and the remaining three options are selected from different emotion groups to ensure that the choices for the QA candidates are selected from each of the groups. The example in Figure 1-1 shows that the correct answer is Surprise from the Pleasure group and the remaining candidates, i.e., Confidence, Anger, and Embarrassment, are drawn from the other groups, i.e., Happiness, Anger, and Sadness, respectively.

STEP2: Generate the Question by VideoLLM Since BoLD does not follow the QA format, we create the appropriate questions from the pairs of candidates and a video by following the prompt design of mCSQA (Sakai et al., 2024a) modified for VideoLLMs. We input the four candidate options (e.g., Confidence, Surprise, Anger, Embarrassment from each group in Figure 1-1) along with the video into Gemini with the highest performance (Team et al., 2024) and let the model generate questions

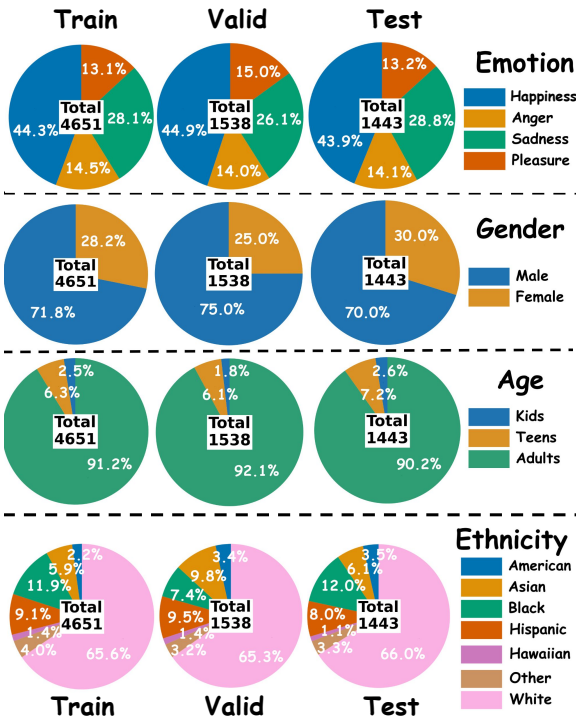


Figure 3: The proportion of metadata in the BQA.

such as “What emotion does the man in the video appear to be exhibiting?” like Figure 1-2 with the prompt we use for the generation in Appendix C.3.

STEP3: Filter the QA by VideoLLM Sometimes, a VideoLLM generates a question which is relatively easy to estimate or answer, e.g., a question that already contains information about the correct candidate, such as “The man looks so shocked. Which emotion is appropriate at this time?” Thus, we let Gemini evaluate whether the generated questions were enough objective and whether they contained superficial information about the correct candidate. If any outputs included the harmful content or did not conform to the conditions for the questions, they were excluded as shown in Figure 1-3.

STEP4: Classify the QA as Easy or Hard Finally, as shown in Figure 1-4, we let Gemini solve the created questions by labeling each question as “Easy” when it could answer or as “Hard” if it could not answer using the prompt presented in Appendix C.3. These question labels allow us to analyze whether a hard question that is not solvable by Gemini is also difficult for other VideoLLMs. For instance, if the correct answer is “Surprise” and Gemini responds with “Anger,” then that question would be classified as “Hard.” After completing these four steps, we split the dataset into training, validation, and test sets in a 6:2:2 ratio, resulting in 4.5k, 1.5k, and 1.5k questions, respectively. The precise number of data is in Table 3.

What are the Aspects that Make the Samples Easy/Hard? Our analysis of the issues labeled as "Hard" revealed the following findings:

(1) Neutral expressions: Videos featuring neutral facial expressions often resulted in the hard label.

(2) Obstructions such as glasses, hats, or sunglasses: Samples involving individuals wearing such items were more likely to be misclassified, leading to the hard label. These findings suggest that VideoLLMs heavily rely on facial expressions when interpreting body language from the footage.

Dataset Analysis Figure 3 shows the distribution of our datasets categorized by our emotion type and three groups of meta information in BoLD: gender, age and ethnicity annotated by humans, revealing that many of the videos feature adult males who are White. It displays the distribution of the four groups of emotions when the annotators selected the most appropriate emotion from the 26 patterns as the correct answer. While Happiness occupies a lot, the overall distribution appears to be balanced.

4 Evaluation

Experimental Setup The models used for evaluation include VideoLLaMA2 (Cheng et al., 2024), LLaVA-NeXT (Zhang et al., 2024a), Qwen2-VL (Wang et al., 2024), and Phi-3.5 (Abdin et al., 2024) with the prompt in Appendix B.2. For the test data, we also used the proprietary models, Gemini (Team et al., 2024) and GPT-4o (OpenAI et al., 2024). We used LoRA-Tuning (Hu et al., 2022) on VideoLLaMA2 with the configuration in Appendix B.1 using the training data and let the model answer with the correct choice in a single word. All audio from the videos was removed to allow for evaluating the model’s ability to interpret body language without relying on auditory information. Additionally, we randomly selected 100 cases from the test set to measure human performance, describing the guideline in Appendix C.1. As an additional evaluation on the test set, we further employed Multimodal Chain of Thought (CoT) to make the models generate reasoning as evidence we call “rationale” for selecting answers. The prompts used for these evaluations are provided in Appendix C.3.

Main Results We show the results in Table 1 with the test set and Table 4 with the valid set. GPT-4o and Gemini achieved higher accuracy than

VideoLLM	#F	Test			Test (CoT)		
		Easy	Hard	Total	Easy	Hard	Total
Human (Rand100)	-	0.96	0.77	0.85	-	-	-
Gemini	*	0.91	0.08	**0.61**	0.94	0.90	0.92
GPT-4o	*	0.78	0.38	0.60	0.97	0.95	0.96
Phi-3.5	16	0.77	0.41	0.58	1.00	0.96	**0.98**
Qwen2-VL	16	0.68	0.27	0.47	0.98	0.95	0.97
LLaVA-NeXT	16	0.66	0.30	0.47	1.00	0.96	**0.98**
VideoLLaMA2	16	0.15	0.01	0.08	0.15	0.01	0.08
VideoLLaMA2 (FT)	16	0.98	0.91	0.94	0.98	0.90	0.94

Table 1: The results using BQA. #F indicates the frame. An asterisk (*) signifies 1 fps. (FT) indicates the LoRA-Tuning model. "Human" is an average of 3 annotators.

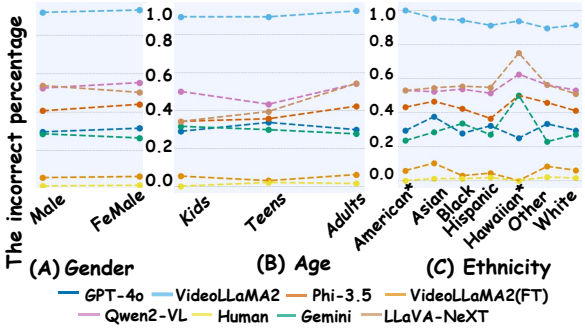


Figure 4: The analysis of incorrectly answered questions shows, from left to right, (A) gender, (B) age, and (C) ethnicity. Note that the higher value indicates more mistakes. An asterisk (*) in (C), especially "American" and "Hawaiian", indicates that they are native humankind.

the other models. VideoLLaMA2 replied without confirming the choice of format before fine-tuning (FT), resulting in a low score, but after FT, its score surpassed Gemini's. From this result, the label assignment in STEP4 did not cause hallucinations. Regarding Gemini, which generated the questions, we found that the problems were sufficiently challenging even for the model itself. Furthermore, in STEP4 of Section 3, those labeled as Easy became unsolvable during inference, likely due to the prompt that restricted the output to single words. Regarding CoT, the results showed a large improvement, confirming its effectiveness for VideoLLMs.

5 Analysis and Discussion

We analyzed the videos by gender (Figure 4-A), age (Figure 4-B), and ethnicity (Figure 4-C) in which each model tends to make mistakes. Other models showed lower evaluation results in the Hard setting compared to the Easy setting. This indicates that even if a language model can create questions, it does not guarantee that it can solve them itself. Since the accuracy of the other VideoLLMs, even GPT-4o, was lower than that of Gemini, the dataset proved to be sufficiently challenging for all models.

Which Age do VideoLLMs Often Mistake? We show which age groups models tend to struggle with in Figure 4-B. Higher values indicate a greater tendency to make errors on videos featuring individuals from that age group. These results show that most models do not exhibit bias based on age, while LLaVA-NeXT tends to make more errors on videos featuring "Adults" compared to the others.

Which Ethnicity do VideoLLMs Often Mistake? In Figure 4-C, we show the consistent patterns to make the mistakes based on the ethnicity of the individuals in the videos. Gemini and LLaVA-NeXT tend to make more errors on the problems related to "Native Hawaiian." Notably, LLaVA-NeXT only achieves around 25% accuracy on these questions.

Why does CoT Improve the Performance? While the Multimodal Chain of Thought (CoT) improved performance, we observed that the rationales often included the correct answer. The inclusion of leaked answers contributes to the performance improvements largely. Thus, the scores achieved by Multimodal CoT should be treated separately from the inherent difficulty of the dataset. However, upon analyzing the types of errors even after CoT, we observed a recurring trend: many of the misclassified instances featured neutral facial expressions or minimal body movements. This observation further supports the claim in Section 3 that VideoLLMs tend to focus more on the face when attempting to understand the body language.

6 Conclusion

Our work created a dataset called BQA to evaluate whether VideoLLMs understand body language that represents emotions and let VideoLLMs solve the questions using BQA. The results show the questions are challenging for all models, confirming the meaning of the dataset. We analyze the types of questions each model tends to get wrong, revealing that some models show a tendency to make more mistakes based on ethnicity and age group. The models also tend to focus more on the face than on body language, and we found that accuracy decreases when there are obstructions that obscure the face. It is also essential to conduct evaluations that focus on the biases in VideoLLMs.

7 Limitations

7.1 The Evaluation by Human Annotator

The random sampling of 100 evaluations was conducted by three people. The overall agreement score is 0.79, indicating a high level of agreement. In the future, we may take care of using crowdsourcing to gather evaluations from a lot of people. Furthermore, it would be beneficial to include evaluations from people of various ethnicities to explore differing perspectives on body language that expresses emotions. However, Tedeschi et al. (2023) argues that human baselines may be unreliable due to factors such as crowdsourced worker payment issues and random sample effects. We should therefore be cautious about the baseline for our human evaluation.

7.2 Video Quality

This study takes care of the possibility that video quality or size may affect accuracy. Specifically, BoLD uses old films, and some of them have noticeably poor quality. Since the same data is used for evaluation, the ranking of the models' accuracy will likely remain unchanged. However, when inputting higher-quality videos, accuracy might improve across all models.

7.3 Frame Issues

Although VideoLLMs claim to handle videos, many actually use image models by treating videos as a sequence of images. In this study, we standardized the number of frames each model can process to 16, but inputting the maximum number of frames may affect the results. However, since inputting more frames increases memory usage, we also need to be mindful of resource constraints.

7.4 Regarding Emotional Expressions

In this study, we categorized 26 patterns of emotional expression into 4 groups based on previous research (James, 1890). While we extracted options from these four patterns, this method may not be entirely accurate. Future research will focus on how the models behave when we expand the available options.

7.5 The Costs of Calling API

The models used in this paper are GPT-4o (gpt-4o-0806) from OpenAI. GPT-4o is accessed via API, which is subject to change and incurs costs based on the number of input tokens. In this study, inference costs totaled approximately $154 and $100 for Multimodal CoT, but this may change in the future. Additionally, due to cost considerations, we used Gemini-1.5-pro. This model is also accessed via API, which is subject to change and incurs costs based on the number of input tokens.

8 Ethical Considerations

8.1 Taking Care about Culture

The expression of emotions through body language doesn't necessarily remain consistent across all countries. Therefore, we might need to update the dataset to take care of cultural factors in future developments.

8.2 License

Since BoLD does not have a clear license, we believe its use for research purposes is unproblematic.

8.3 Large Scale Human Evaluation

Although human evaluations of BQA aim to minimize bias, implicit biases may still remain. In the future, it may be necessary to employ multiple annotators for a fair assessment. However, as mentioned in Tedeschi et al. (2023), careful consideration is needed when hiring annotators, as their results may not always be accurate.

8.4 AI Assistant Tools

We used ChatGPT[2] and DeepL[3] to translate sentences to English to accelerate our research.

8.5 Annotators in BoLD

In this study, we rely on data annotated in BoLD for analysis. However, the annotated information may not always be accurate. For example, a White annotator may have intentionally mislabeled an Asian person as Black. Additionally, implicit biases from annotators could lead to adults being mistaken for children.

Regarding emotions, there is also a possibility of bias during the annotation process. We ourselves found it challenging to explain the differences between the 26 patterns of emotional expression, which is why we grouped them into four categories. It is unlikely that annotators fully captured these distinctions, so we must approach this with caution.

[2] https://chatgpt.com/

[3] https://www.deepl.com/ja/translator

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775, Bangkok, Thailand. Association for Computational Linguistics.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. 2024. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *Preprint*, arXiv:2406.07476.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *arXiv preprint arXiv:1809.02922*.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Rongrong Ji, and Xing Sun. 2024. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *Preprint*, arXiv:2405.21075.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Lee Hyun, Kim Sung-Bin, Seungju Han, Youngjae Yu, and Tae-Hyun Oh. 2024. SMILE: Multimodal dataset for understanding laughter in video with language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1149–1167, Mexico City, Mexico. Association for Computational Linguistics.

William James. 1890. *The Principles of Psychology*. Dover Publications, London, England.

Ronak Kosti, Jose M. Alvarez, Adria Recasens, and Agata Lapedriza. 2017. Emotion recognition in context. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1960–1968.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. 2018. TVQA: Localized, compositional video question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1379, Brussels, Belgium. Association for Computational Linguistics.

Yu Luo, Jianbo Ye, Reginald B. Adams, Jia Li, Michelle G. Newman, and James Z. Wang. 2020. Arbee: Towards automated recognition of bodily expression of emotion in the wild. *International Journal of Computer Vision*, 128(1):1–25.

Teresa Lynn, Malik H. Altakrori, Samar M. Magdy, Rocktim Jyoti Das, Chenyang Lyu, Mohamed Nasr, Younes Samih, Kirill Chirkunov, Alham Fikri Aji, Preslav Nakov, Shantanu Godbole, Salim Roukos, Radu Florian, and Nizar Habash. 2025. From multiple-choice to extractive QA: A case study for English and Arabic. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2456–2477, Abu Dhabi, UAE. Association for Computational Linguistics.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. 2024. Video-ChatGPT: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12585–12602, Bangkok, Thailand. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. LLM Evaluators Recognize and Favor Their Own Generations. In *Advances in Neural Information Processing Systems*, volume 37, pages 68772–68802. Curran Associates, Inc.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Yusuke Sakai, Hidetaka Kamigaito, and Taro Watanabe. 2024a. mCSQA: Multilingual commonsense reasoning dataset with unified creation strategy by language models and humans. In *Findings of the Association for Computational Linguistics ACL 2024*,

pages 14182–14214, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Yusuke Sakai, Adam Nohejl, Jiangnan Hang, Hidetaka Kamigaito, and Taro Watanabe. 2024b. Toward the evaluation of large language models considering score variance across instruction templates. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 499–529, Miami, Florida, US. Association for Computational Linguistics.

Haruki Sakajo, Yusuke Sakai, Hidetaka Kamigaito, and Taro Watanabe. 2025. Tonguescape: Exploring language models understanding of vowel articulation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12605–12619, Albuquerque, New Mexico. Association for Computational Linguistics.

Jaehyung Seo, Jaewook Lee, Chanjun Park, SeongTae Hong, Seungjun Lee, and Heuiseok Lim. 2024. KoCommonGEN v2: A benchmark for navigating Korean commonsense reasoning challenges in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2390–2415, Bangkok, Thailand. Association for Computational Linguistics.

Jaehyung Seo, Seounghoon Lee, Chanjun Park, Yoonna Jang, Hyeonseok Moon, Sugyeong Eo, Seonmin Koo, and Heuiseok Lim. 2022. A dog is passing over the jet? a text-generation dataset for Korean commonsense reasoning and evaluation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2233–2249, Seattle, United States. Association for Computational Linguistics.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, and 1118 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Simone Tedeschi, Johan Bos, Thierry Declerck, Jan Hajič, Daniel Hershcovich, Eduard Hovy, Alexander Koller, Simon Krek, Steven Schockaert, Rico Sennrich, Ekaterina Shutova, and Roberto Navigli. 2023. What's the meaning of superhuman performance in today's NLU? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12471–12491, Toronto, Canada. Association for Computational Linguistics.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *Preprint*, arXiv:2409.12191.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. NExT-QA: Next Phase of Question-Answering to Explaining Temporal Actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9777–9786.

Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2025. mPLUG-owl3: Towards long image-sequence understanding in multi-modal large language models. In *The Thirteenth International Conference on Learning Representations*.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. 2020. Clevrer: Collision events for video representation and reasoning. In *International Conference on Learning Representations*.

Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. 2024a. Llava-next: A strong zero-shot video understanding model.

Zhuosheng Zhang, Aston Zhang, Mu Li, hai zhao, George Karypis, and Alex Smola. 2024b. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models.

In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 1097–1100, New York, NY, USA. Association for Computing Machinery.

A Additional Discussions

A.1 Which Emotion Do VideoLLMs Often Mistake?

We analyzed which labels the models predicted for the questions they answered incorrectly in Figure 5. The x-axis represents the correct labels, while the y-axis shows the emotions predicted by the models. The "Others" includes instances where the model output was not an emotion, such as a sentence. None of the models predicted "Pleasure" when they made mistakes, indicating a strong capability to predict actions representing "Pleasure." However, all models frequently failed when the correct label was "Happiness," often selecting opposing emotions like "Sadness" or "Anger."

A.2 Expansion from existing QA to Other QA Datasets

There are several methods for extending existing QA datasets to new tasks or modalities. For example, Seo et al. (2024) created KoCommonGEN v2 as an expansion of Korean CommonGen (Seo et al., 2022), and Lynn et al. (2025) constructed a new culturally-aware dataset based on Belebele (Bandarkar et al., 2024). Demszky et al. (2018) transformed existing QA datasets, such as SQuAD (Rajpurkar et al., 2016) and MovieQA (Tapaswi et al., 2016), into an NLI dataset, highlighting how adapting existing resources can yield valuable contributions. Similarly, Sakajo et al. (2025) developed Tonguescape to benchmark VideoLLMs. Building on this trend, we extend BoLD to construct BQA, a dataset designed to serve as input for VideoLLMs and support the understanding of body language.

A.3 Which Gender do VideoLLMs Often Mistake?

Figure 4-A shows the tendency of each model to make mistakes based on whether the video features a male or female subject. Higher values indicate a higher likelihood of errors for the videos. From these results, we can see that none of the models exhibits bias based on gender. These findings suggest that the models are focused on human actions rather than whether the person is male or female.

A.4 Trends in Generated Questions

As described in Section 3, we used Gemini, which achieved the best performance among VideoLLMs according to (Fu et al., 2024), to generate the question texts. We confirmed that the generated questions predominantly begin with wh-words, i.e., `What`, `Where`, `When`, `Who`, `Why`, `How`, `Which`, which we believe is appropriate for question formation. Furthermore, we evaluated the diversity of the generated questions using Self-BLEU (Zhu et al., 2018), and the results are shown in Table 2. These results are based on 4-gram analysis[4]. They indicate that the diversity of the questions is balanced across sets, confirming the fairness of the question distribution.

Set	Score (↓)	n-gram
Train	0.87	4
Valid	0.86	4
Test	0.85	4

Table 2: The results of Self-BLEU on BQA. Lower scores indicate the more diverse questions.

A.5 Data Contamination

We do not believe that BQA data has already been used for training for the following reasons.

(1): Originality of the questions: The questions used in this study were generated by the model itself during the experimental process. Currently, there is no scenario where the specific question-and-answer pairs created in this study would have been included in the pre-training data of proprietary models like Gemini or GPT-4o.

(2): Performance inconsistency with contamination: If data contamination had occurred, one would expect Gemini to have a significant advantage in answering the questions, as it would have been exposed to similar patterns during pre-training. However, as the results show, Gemini achieves only approximately 60% accuracy on the dataset. This relatively modest performance strongly suggests that no data leakage has occurred. If contamination were present, the accuracy would likely be substantially higher, especially for questions generated by Gemini itself.

[4]When calculating the score, we use nltk (Bird and Loper, 2004).

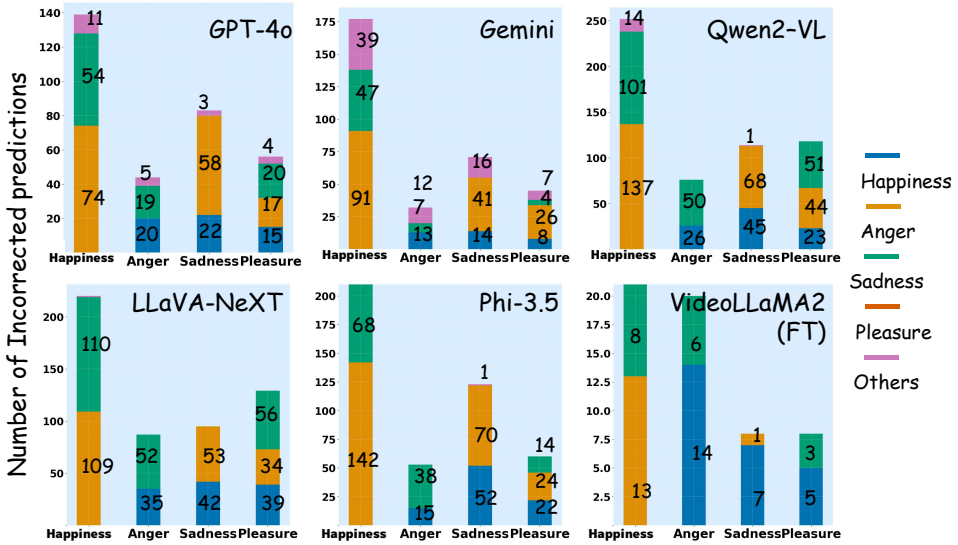


Figure 5: The emotional distribution of output from each model. The x-axis shows the label of the correct answer, and the y-axis shows how the model got it wrong in doing so.

A.6 Self-Preference Bias

Self-preference bias is a phenomenon where a Large Language Model (LLM) favors its own outputs over responses from other models or human-generated texts (Panickssery et al., 2024). This bias can compromise evaluation objectivity, making it essential to address its potential impact on our results. However, in this study, only the questions were generated by the Gemini model. The answer candidates were sourced from external datasets, i.e., BoLD, ensuring they were independent of Gemini's generation process. This approach prevents any alignment-based advantage from self-generated outputs, mitigating the risk of self-preference bias.

A.7 Exhortation of Multimodal CoT

We analyzed the large accuracy improvements achieved by applying Multimodal Chain of Thought (Zhang et al., 2024b). Upon examining the generated rationales, we observed that many explanations explicitly included reasons for selecting the correct answer while also providing reasons for not choosing the other options. Multimodal CoT presents the question, the answer choices, and the correct answer, then generates an explanatory rationale that derives the answer from the given choices, this rationale being subsequently used for the QA task. These rationales often clearly incorporated the correct answer within the explanation itself, as we described below the boxes. Unlike the traditional step-by-step reasoning approach of Chain of Thought (Wei et al., 2022), this explicit inclusion of the correct answer suggests a form of answer leakage, which naturally contributes to higher accuracy. Therefore, it is important to note that this answer leakage influences the results obtained from Multimodal CoT and may not fully reflect the inherent capabilities of the model.

Examples of Successful Generation about Rationale

The man sitting in the armchair makes eye contact with the other man and sustains it while he speaks. This direct eye contact and attention to the conversation suggest active participation and engagement with what the other man is saying.

Examples of Bad Generation about Rationale (Data Leakage)

The speaker is looking for an address that she thought she had placed in her purse. When she realizes it is not there, she laughs and appears flustered, indicating confusion and a lack of certainty. This best aligns with the "doubt_confusion" emotion category. The other choices are not applicable because she is not displaying affection, aversion, or disconnection.

A.8 Dataset Size

The size of the BQA dataset after applying the four steps described in Section 3 is in Table 3. Ensuring that the dataset is sufficient to evaluate whether VideoLLMs can comprehend body language.

A.9 The Result of Valid Set in BQA

Table 4 below presents the experimental results obtained using the validation set. We evaluated

Split	Total Size	Easy	Hard
Train	4,651	2,192	2,459
Valid	1,538	746	792
Test	1,443	707	736

Table 3: The data size of each split. After completing the 4 steps of dataset creation, we split the data. Our work also conducted LoRA-Tuning.

open models using the validation set as part of the evaluation.

VideoLLM	#F	Valid		
		Easy	Hard	Total
Phi-3.5	16	0.76	0.38	**0.56**
Qwen2-VL	16	0.69	0.32	0.50
LLaVA-NeXT	16	0.65	0.31	0.47
VideoLLaMA2	16	0.40	0.09	0.24
VideoLLaMA2 (FT)	16	0.89	0.68	0.78

Table 4: The result of valid set, #F indicating the frame.

B Details of Experimental Settings

Below, we described the details of the models evaluated in this study.

Model	Params	HuggingFace Name
Qwen2-VL	8.29B	Qwen/Qwen2-VL-7B-Instruct
LLaVA-NeXT	8.03B	lmms-lab/LLaVA-NeXT-Video-7B-Qwen2
Phi-3.5	4.15B	microsoft/Phi-3.5-vision-instruct
VideoLLaMA2	8.03B	DAMO-NLP-SG/VideoLLaMA2-7B-Base
GPT-4o	-	gpt-4o-2024-0806
Gemini	-	gemini-1.5-pro

B.1 LoRA Tuning Setting

We conducted LoRA (Hu et al., 2022) tuning with the VideoLLaMA2 model. The model was trained using four NVIDIA A100-SXM4-40GB GPUs. Detailed parameters are provided in Table 5.

B.2 Gemini Settings

Why we use Gemini for question generation stems from its state-of-the-art performance at the time of this study. According to Fu et al. (2024), as of later in 2024, Gemini demonstrated the highest performance among multi-modal LLMs in video QA tasks. Consequently, Gemini was selected for question generation, as it represented the most advanced model available during the research period.

Table 6 describes the configuration used to let the Gemini inference and generate in this study.

Hyper Parameter	Value
torch_dtype	bfloat16
seed	42
max length	2048
batch size	4
epoch	1
lora r	128
lora alpha	256
lora dropout	0.05
lora target modules	o_proj, gate_proj, up_proj, v_proj, q_proj, down_proj, k_proj

Table 5: The hyperparameters of VideoLLaMA2 (Cheng et al., 2024) for LoRA-Tuning (Hu et al., 2022) used in the experiment, and others, were set to default settings (Sakai et al., 2024b). The implementation used the Transformers library (Wolf et al., 2020).

Category	Value
HARM_CATEGORY_DANGEROUS HARM_CATEGORY_HARASSMENT HARM_CATEGORY_HATE_SPEECH HARM_CATEGORY_SEXUALLY_EXPLICIT HARM_CATEGORY_DANGEROUS_CONTENT	BLOCK_NONE

Table 6: The configuration settings of Gemini.

B.3 Filtering by Rule-Based Algorithm

In this study, we first performed a rule-based filtering process. We checked if the questions ended with a question mark, whether they were a single line, and ensured that the candidates for the options were not included in the questions. However, no instances were excluded during this filtering.

B.4 The Proportion of Data Split

During the creation phase of BQA, if Gemini determined that a question that Gemini created was harmful, we removed it from the dataset. We then had Gemini attempt to solve the created questions, labeling them as "Easy" and "Hard". Figure 6 shows the distribution of those labels.

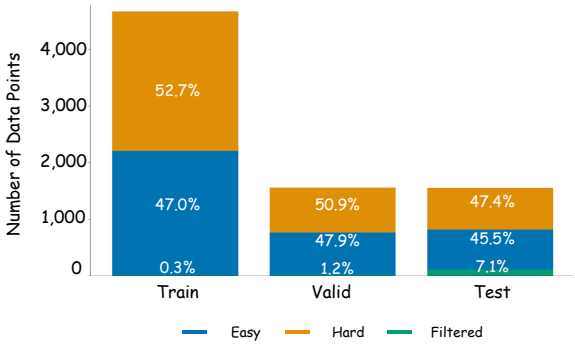


Figure 6: The percentage of each data. As stated in Section 3, we filtered the problem statements in STEP3 and set the difficulty levels of the problems in STEP4.

C Instruction and the Prompt

C.1 Instruction for Human Evaluation

We conducted human evaluations of the BQA test data. The instruction for requesting humans is shown below.

The Instruction for Human Evaluation

```
Your task is to watch videos related to body language, read the questions, and select the
appropriate option.
Please be maximally careful about bias and try to evaluate appropriately.
Please answer with only the word of the chosen option. There are 100 questions, so take breaks
as needed, and focus intently during the evaluation.

Question:
{question}

Choice:
{choice0}
{choice1}
{choice2}
{choice3}
```

C.2 The Content of the Dataset

The structure of the metadata contained in the dataset is outlined below. The complete dataset is released at https://huggingface.co/datasets/naist-nlp/BQA.

An Example of Dataset

```
{
    "video_url": "URL_TO_VIDEO",
    "answer": "affection",
    "question": "What feeling does the man in the video seem to express when
he is smiling?",
    "prompt": "Please look at the video entered and choose the option that
applies to the following question statement.

        Question:
        What feeling does the man in the video seem to express when he is smiling?

        Choice:
        affection
        sympathy
        aversion
        doubt_confusion


        Please output only the words that apply to your answer.
        If you output sentences or symbols, the answer will be incorrect.",
    "easy_hard_label": "hard"
}
```

C.3 The Prompt on Creating a Dataset

Below, we present the prompts used to instruct the model while creating the BQA. One prompt was for generating questions from the video and the candidates, and the other was for filtering the generated questions to determine whether they adhered to the specified conditions.

STEP2: The Prompt for Generate the Question

```
First, please understand the video.
Understanding the video, please consider the candidates entered and the correct answer, and
create a question for which the correct answer is the answer.

candidates:
{candidates[0]}
{candidates[1]}
{candidates[2]}
{candidates[3]}

answer: {answer_emotion}

Notes:
1. Do not include the word choice in the question text.
2. Do not include superficial information such as word counts or paraphrases that can be answered
by imagination.
3. Always end a sentence with a '?'.
4. Attempt to create an objective question.
5. Consist of only one sentence.
```

STEP3: The Prompt for Filtering

```
Determine whether the input question meets the following conditions.
If it is judged to meet all five conditions, return the input sentence as it is. If it does not
meet the conditions, output only the word "HARMFUL" in all uppercase letters.

The output should be either the input sentence or "HARMFUL" only.

Five conditions:

1. Does not use words from multiple-choice options.
2. Avoids using superficial information like character count, etc.
3. Ends with a question mark (?).
4. Is an objective question.
5. Consists of only one sentence.
Input sentence: {question}
```

STEP4: The prompt for adding the Label as Easy/Hard

```
Question:
{question}

Choice:
{choice0}
{choice1}
{choice2}
{choice3}
```

The Prompt for Inference

```
Please look at the video entered and choose the option that applies to the following question
statement.

Question:
{question}

Choice:
{choice0}
{choice1}
{choice2}
{choice3}

Please output only the words that apply to your answer.
If you output sentences or symbols, the answer will be incorrect.
```

The Prompt for generating rationale for Chain of Thought

```
Please look at the video entered and choose the option that applies to the
following question statement.

Question:
{question}

Choice:
{choice0}
{choice1}
{choice2}
{choice3}

Answer: {answer}

Please explain the rationale to choose the correct answer.
Solution:
```

The Prompt for Inference with Chain of Thought

```
Please look at the video entered and choose the option that applies to the
following question statement.

Question:
{question}

Choice:
{choice0}
{choice1}
{choice2}
{choice3}

Rationale:
{cot}

Answer:"""
```

C.4 Examples of BQA

We provide several examples of actual questions and the corresponding answers from each model. The images in the Video column capture the most distinctive moments from the videos (i.e., thumbnails). The complete dataset is available at https://huggingface.co/datasets/naist-nlp/BQA. The prompt includes only the necessary parts; for the complete details, we describe in Appendix C.3.

Input video

Question

Question:
What is the emotion shown by the man toward the people he is talking about?

Choice:
aversion
pleasure
esteem
fear

Prediction

Gemini: **fear**
GPT-4o: **fear**
Phi-3.5: **fear**
Qwen2-VL: **fear**
LLaVA-NeXT: **fear**
VideoLLaMA2: **fear**
VideoLLaMA2(FT): **fear**

Answer

aversion

Figure 7: An example of BQA.

Input video

Question

Question:
What does the man's demeanor suggest he is experiencing?

Choice:
fatigue
engagement
disapproval
surprise

Prediction

Gemini: **disapproval**
GPT-4o: **fatigue**
Phi-3.5: **fatigue**
Qwen2-VL: **disapproval**
LLaVA-NeXT: **disapproval**
VideoLLaMA2: **disapproval**
VideoLLaMA2(FT): **disapproval**

Answer

fatigue

Figure 8: Another example of BQA.

Grounded, or a Good Guesser? A Per-Question Balanced Dataset to Separate Blind from Grounded Models for Embodied Question Answering

Miles Shelton[1] **Nate Wingerd**[1] **Kritim K. Rijal**[1] **Ayush Garg**[1]
Adelina Gutic[1] **Brett Barnes**[1] **Catherine Finegan-Dollak**[2]
University of Richmond Department of Computer Science
[1]{first.last}@richmond.edu
[2] Corresponding Author, cfinegan@richmond.edu

Abstract

Embodied question answering (EQA) means using *perception of* and *action in* an environment to answer natural language questions about that environment. However, previous work has demonstrated that blind language models (which do not incorporate perception, but predict an answer based solely on the question text) are a strong baseline for existing benchmarks, even compared against state-of-the-art vision and language models. To determine whether a model is grounding its answers in its specific environment, rather than relying on a language model's expectations about the world generally, we propose PQB-EQA, a *per-question balanced* EQA dataset. In this new benchmark, every question appears twice, paired with two different environments that yield two different answers. That is, the answer distribution is balanced for each question, not just across the whole dataset. We show both theoretically and empirically that grounding in the environment is necessary to perform better than chance on PQB-EQA.[1]

1 Introduction

Imagine a search and rescue robot that could answer the question, "What is behind the concrete slab?" by navigating to a location where it could find the answer. To answer the question correctly, the agent would need to (1) understand the natural language question, (2) perceive its environment (using, for instance, vision), and (3) select actions to find the necessary information; that is, it would need to engage in *embodied question answering* (EQA) (Das et al., 2018). Obviously, such an agent should ground its answers in the environment; we do not want the search and rescue robot to tell us what is *often* behind concrete slabs. There is therefore a need to test such grounding in EQA models.

Unfortunately, "blind" language models—that is, models that receive only the question text and no images—have been strong baselines for existing EQA benchmarks (Anand et al., 2018; Thomason et al., 2019; Ilinykh et al., 2022; Majumdar et al., 2024), showing that they do not require the model to use perception, let alone to act in its environment to find the answer. Performance on these benchmarks therefore does not tell us whether a multimodal language model is in fact grounding its answers in its environment or hallucinating based on patterns in language or datasets. Because EQA is meant to enable agents to answer questions *about an environment*, answers based solely on language priors are unreliable for real-world use (Thomason et al., 2019; Ilinykh et al., 2022); work in the related area of visual question answering (VQA) further supports this (Zhang et al., 2016; Goyal et al., 2017).

This paper introduces PQB-EQA, a new EQA benchmark with per-question balancing, a dataset design strategy to ensure that a model that does not perceive its environment cannot perform better than chance. Each example is a question-environment-answer tuple. Every question depends on its environment; for example, "Could you tell me if there are cobwebs on the houses to the southwest?" might be answered *yes* in reference to one environment and *no* in reference to another, as shown in Figure 1. PQB-EQA pairs each question with two different environments giving two different correct answers. Only a model that integrates language and perception can determine the correct answer for the given question-environment pair. In addition, human testers demonstrated that action in the environments is necessary to find the answers.

The contributions of this paper are the following:

- We construct the first per-question-balanced benchmark for EQA.
- We demonstrate that a state-of-the-art language model performs no better than chance

[1]Data and code available at https://milesshelton.github.io/pqb_eqa/

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 124–135
July 27 - August 1, 2025

(a) The agent turns to the southwest and quickly spots cobwebs on the sandstone house.

(b) The agent turns and walks into the house to look for cobwebs before determining there are none.

Figure 1: Screenshots collected from humans in two environments answering the question, "Could you tell me if there are cobwebs on the houses to the southwest? A. yes B. no."

on our new benchmark, while adding vision and a set of oracle actions greatly improves performance.

- We verify that humans need to take action in the environments to answer the questions.

2 Background: EQA Task Definition

The EQA task can be understood in terms of its inputs (an environment and a question about it) and outputs (an answer). An agent capable of certain actions is given a natural language question and placed in a partially-observable environment (as a fully-observable environment would obviate the need for actions to find the answer). The agent uses actions to explore the environment to find the answer. EQA specifically addresses questions *about the environment* (distinguishing it from text-only QA) and allows the agent to select actions to obtain information, distinguishing it from VQA.

More formally, we define the task as follows: a model, M, is given a partially-observable environment E, and a natural language question, Q, which is a sequence of words $w_1 ... w_n$ from the input vocabulary, V_i. The goal is to predict the correct answer Y conditioned on E. At each time step t, M receives an observation from E, which we call obs_t. M may choose an action, a_t, from an action space, A. The action at time t may affect the observation at time $t+1$. The goal remains to predict the answer Y in light of the environment E; action choice is a latent variable. Each example in a dataset is therefore a tuple, $\{Q, E, Y\}$.

3 Per-Question Balancing

Dataset balancing is a technique to prevent models that learn spurious correlations from appearing better than they are. If the answer "yes" shows up much more frequently than "no" in a dataset, a system that picks the majority answer will be a strong baseline without requiring language or vision skills. Some EQA datasets addressed this problem in their datasets by balancing their answers. For example, Tan et al. (2023) generated questions and sampled so that their final dataset contained precisely the same number of "Yes" answers and "No" answers, as well as equal numbers of counting questions answered 0, 1, 2, 3, or 4. This prevents a model from exploiting dataset-wide patterns in the answers.

However, balancing answers across the entire dataset is insufficient to prevent blind language models from falsely appearing to be effective. Consider an extreme example: Suppose a dataset had environments that always contained dogs but never contained cats. By sampling an equal number of questions that asked if there were dogs and questions that asked if there were cats, one could easily create a dataset that was perfectly balanced across answers but would not require a model to integrate language and vision, since language alone would be enough to answer the questions correctly.

Per-question dataset balancing, by contrast, addresses the blind language model problem by balancing the distribution of correct answers *for each question*. This means that every question must appear in multiple $\{Q, E, Y\}$ tuples; that is, for a given question q_i, there must exist two tuples, $\{q_i, e_i, y_i\}$ and $\{q_i, e'_i, y'_i\}$, such that $e_i \neq e'_i$ and $y_i \neq y'_i$. Since every question has two distinct answers, recognizing the type of question—or indeed, the exact question asked—cannot be enough for a model to guess the correct answer. Although this technique has been used for static visual question answering (VQA) datasets (Hodosh and Hockenmaier, 2016; Zhang et al., 2016; Goyal et al., 2017), it has not previously been applied to EQA.

4 Dataset

We constructed a per-question balanced dataset comprised of 424 question-environment-answer tuples and recordings of human players.

4.1 Environments

Environments in the dataset are bounded sections of a world in the video game Minecraft. Minecraft has been a popular test bed for reinforcement learning (Guss et al., 2019; Johnson et al., 2016; Kiseleva et al., 2021; Baker et al., 2022; Fan et al., 2022; Wang et al., 2023) and grounded natural language processing (Kitaev and Klein, 2017; Szlam et al., 2019; Narayan-Chen et al., 2019; Srinet et al., 2020; Bonn et al., 2020; Jayannavar et al., 2020; Burns et al., 2021; Kiseleva et al., 2021; Shi et al., 2022), although it has not previously been used for EQA. Due to its highly engaged online community, an abundance of text, images, and videos related to the game are available for training (Fan et al., 2022).

There is a tradeoff between realism and variety of environments for EQA. Previous EQA datasets sought to use the most realistic possible visual simulations, which limited their environments to houses and offices (Gordon et al., 2018; Das et al., 2018; Ren et al., 2024; Majumdar et al., 2024). In contrast, using Minecraft as our simulation enables a wide range of environment types. Minecraft natively incorporates 64 different biome types, such as deserts and villages, and over 1,000 distinct items, from gold to mushroom stew. It also encourages imaginative construction, allowing new objects to be added. Since prior datasets for EQA have already provided realism, we contribute environments that enable greater variability.

Using the WorldEdit Minecraft mod,[2] we constructed 312 60-by-60 block, single-biome environments. These include city, town, cave, desert, mansion, nether, plains, and snow environments with variations in the items, structures, and creatures placed in each. All environments are compatible with the MineRL (Guss et al., 2019) v.1.0 framework for training RL agents, enabling EQA to benefit from existing action-selection modules.

4.2 Questions

To ensure truly natural language, all questions were collected from and curated by humans. For question collection, we recorded dialog between paid human participants playing cooperative minigames where their scores depended on accurately gathering information. Details of the procedures are in Appendix A. We manually identified questions from the game dialogs that (a) depend on the environment (that is, the answer will be different in different environments), (b) are possible for the agent to answer without damaging the environment, and (c) do not require dialog context beyond the question. Some questions immediately met the criteria; others we modified to meet the criteria (for example, we replaced pronouns with their antecedents from the dialog context). See Appendix B for details of question curation.

4.3 Answers

To achieve per-question balancing, we matched each question to two environments that yielded different answers, using a mix of automation and manual review. Using information from the Minecraft save file about items in the environment, we can programmatically determine the answers to some questions; for instance, "are there pigs?" is a lookup to see if any entities in the environment are pigs. However, other questions are more complex; for instance, "are all rooms the exact same?" is hard to automate when the specification lists one structure as "house," not multiple structures as "rooms." We manually found two environments with different answers for these questions. We used the two answers for each question, each correct in a different environment, to construct a multiple choice question.

Answer types are varied. Approximately half are yes-or-no, but 30% of answers appear only once in the dataset. For examples of questions and answer choices, see Appendix C.

To ensure tuple quality, we played through the environments and answered the questions. If the reviewer's answer matched the annotated answer, we kept the tuple; otherwise, a second reviewer answered the questions, and we kept the majority label. After review, we removed all questions that had the same answer in both environments.

4.4 Human Play Data

We recorded human players navigating the environments to answer the questions during tuple review. Once per second, we saved a screenshot. Once per in-game tick (1/20th of a second), we logged all actions taken by the user during that tick. The action space includes actions like "forward," "jump", "left", "right", and "camera." Camera movement is controlled by mouse; the other actions are controlled with keypresses. Ongoing actions, such as forward movement that lasted more than a single tick, appear in each tick's log until they end.

[2] https://github.com/EngineHub/WorldEdit

Model	Acc.	p
Blind GPT4-o	0.507	$p = 0.8082$
GPT4-o with Vision	0.827	$p < 0.0001$

Table 1: Accuracy and p of the two models. The blind model does not significantly outperform chance, while the grounded model does.

Benchmark (Reported in)	Δ Perf.
EQA v1 (Ilinykh et al., 2022)	1.8
A-EQA (Majumdar et al., 2024)	6.3
PQB-EQA *(ours)*	**32.0**

Table 2: Reported difference in scores between models with and without vision on previous benchmarks and on PQB-EQA.

	Yes/No	Other
Blind GPT4-o	0.509	0.505
GPT4-o with Vision	0.796	0.861

Table 3: Results of each model on yes/no questions and all other types of questions. The model with vision outperforms the blind model by a wide margin on both categories of question.

5 Experiments

We conducted experiments to verify that (a) a blind model would *not* perform better than chance, and (b) a model that successfully navigated and perceived its environment *would* perform better than chance. Both elements are needed to ensure that the benchmark cannot be gamed by a blind model but will fairly evaluate truly grounded models.

Text-only GPT4-o was our blind model. We provided it with the questions and answer choices described in Sections 4.2 and 4.3 and instructed it to choose the best answer and guess if it is unsure. Prompt details are in Appendix D. This is a state-of-the-art large language model, but if our hypothesis is correct, it will not outperform chance, since it does not observe the environment.

To verify the dataset's feasibility for a model that perceives and acts in its environment, we provided sequences of oracle screenshots from the human play recordings to GPT4-o with vision.[3] We adjusted the prompt to refer to the images, but kept it otherwise the same as blind GPT4-o. If our hypothesis is correct, this model should achieve significantly better than random accuracy.

We evaluated accuracy and measured significance using a two-tailed binomial test.

6 Results and Discussion

Table 1 summarizes results. Blind GPT4-o achieved accuracy of 50.7%, which is not significantly different from chance ($p = 0.81$). GPT4-o with oracle screenshot sequences achieved accuracy of 82.7%, significantly outperforming chance ($p < 0.0001$). We therefore conclude that the dataset is feasible for grounded EQA models but not blind models.

Works comparing blind models against models with vision on previous benchmarks have observed notably smaller differences, as summarized in Table 2. Ilinykh et al. (2022)[4] reported that on EQA v1, their language-only model achieved accuracy of 36.2, while their vision + language model achieved accuracy of 38.0, a difference of 1.8 percentage points. Majumdar et al. (2024) reported that on their A-EQA dataset, blind GPT-4 achieved an LLM-Match score of 41.8, while GPT4-V achieved 35.5, a difference of 6.3 percentage points. On our dataset, the difference in accuracy is 32 percentage points.

To verify that the difference in performance holds across question types, we split our dataset into yes/no questions and all other types. Based on the results in Table 3, we conclude that the findings hold over different question types.

To verify that the dataset is suited to EQA and not simply static VQA, we analyzed the human action logs. On average, humans took 278.8 actions to answer a question. Less than 25% of recordings included under 50 actions, and only three did not require action. We therefore conclude that in order for an agent to succeed at this task when it is not given the oracle screenshots, it will need to integrate language, perception, *and* action.

7 Related work

In recent years, many EQA datasets have been released, beginning with Gordon et al. (2018)'s in-

[3]For these experiments, we use an oracle sequence of actions, as our goal is not to evaluate an EQA model, but rather to verify qualities of the benchmark.

[4]The reported model was not GPT-4, so some caution should be used in comparing their results to ours; however, running new experiments using GPT-4 on the EQA v1 benchmark is infeasible due to its dependence on currently unavailable SUNCG environments.

teractive question answering (IQA) and Das et al. (2018)'s EQA v1. Variations on the EQA task have focused on multi-target EQA (Yu et al., 2019), EQA with fine-grained robotic manipulation (Deng et al., 2021), EQA with a knowledge base (Tan et al., 2023), EQA with "situational" questions (Dorbala et al., 2024), EQA in a photo-realistic simulation (Ren et al., 2024), and "episodic memory" EQA (Majumdar et al., 2024).

Several EQA datasets are balanced in some way, but none are balanced per question. Das et al. (2018) excluded questions with low normalized entropy for the answer distribution; however, the resulting dataset was still susceptible to blind models (Anand et al., 2018; Thomason et al., 2019; Ilinykh et al., 2022). Yu et al. (2019) applied the same technique with a higher entropy threshold. Deng et al. (2021) manipulated the distribution of answers to be uneven in ways based on anticipated practical applications. Tan et al. (2023) balanced answers across the dataset by over-generating questions and answers and sampling to get equal numbers of "yes" and "no" answers and a uniform distribution over the numbers 0-4 for counting questions. As noted (Section 3), this form of balancing could leave more subtle patterns in the data that a blind model can still exploit. The closest to a per-question balanced dataset for EQA is IQUAD v1 (Gordon et al., 2018), which associated each question with multiple environments; however, an analysis of that dataset shows that only 5 out of 128 questions are equally likely to be true or false.

Previous work on per-question balancing (Hodosh and Hockenmaier, 2016; Zhang et al., 2016; Goyal et al., 2017) applied to visual question answering (VQA) for static images. It does not incorporate environments that agents may act in. Such environments present additional challenges; for example, an EQA agent may have to choose different actions to answer the same question in two different environments, as in Figure 1.

8 Conclusion

This work presented the first per-question balanced dataset for embodied question answering. Per-question balancing ensures that ungrounded models cannot perform well. We gathered natural language questions from human participants and aligned each one with two different environments yielding two different answers. Experiments with GPT-4o verify that even a state-of-the-art hyper-LLM performs no better than chance when using only language, and that the task would be feasible for a model that successfully navigated its environment to find the visual inputs needed to answer the questions. Thanks to its compatibility with MineRL, PQB-EQA is suited for experiments that test the integration of all three parts of the EQA task: language, perception, and action. Future work should evaluate models other than GPT-4 on this benchmark to provide a wider variety of baselines. In addition, we plan to evaluate using end-to-end EQA systems where an agent chooses the actions in place of the oracle action selection used here.

9 Limitations

Embodied question answering has three components: natural language understanding, perception, and action. Our evaluation uses oracle action sequences; future work should incorporate action selection and the perception generated from the selected actions.

Because the questions are multiple-choice, this dataset is not suitable for evaluating generation of answers. This limitation was necessary to ensure per-question answer balancing. Where generation of answers and ensuring that answers are grounded are both important, this dataset should be used in combination with other evaluations.

The dataset is only in English.

Environments are not photorealistic; however, we believe that the greater breadth of environments this enables balances this limitation.

The present dataset's size makes it suitable for evaluation, but it is not large enough to train new models.

10 Ethics

All EQA-based technology carries the risk of misuse for purposes such as inappropriate surveillance by bad actors. In addition, EQA systems that are used for ethical applications but provide inaccurate information may cause harm.

The questions in PQB-EQA were collected from paid human participants aged at least 18 years. Our data collection procedure was reviewed by our institution's IRB and considered to pose no more than minimal risk to participants. Participants were compensated $15 per hour, exceeding the local minimum wage. Informed consent was obtained from all participants. They were made aware that the data collected would be made publicly avail-

able for research. The nature of the data collection (asking questions about a Minecraft game) made disclosure of personally identifiable information unlikely; however, we manually reviewed the results and deleted the sole instance where a participant's name appeared in a question.

Acknowledgments

We thank the anonymous reviewers for their thoughtful feedback, which helped improve the paper. We are grateful to Laura Biester for her comments on early drafts and to Amerah Rutherford for her assistance with testing data collection instructions and performing quality assurance on code. Some of the authors were supported by University of Richmond Arts & Sciences Summer Fellowships. This work was also supported in part by a grant from the University of Richmond Undergraduate Research Committee.

References

Ankesh Anand, Eugene Belilovsky, Kyle Kastner, Hugo Larochelle, and Aaron Courville. 2018. Blindfold Baselines for Embodied QA. ArXiv:1811.05013 [cs].

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. 2022. Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. ArXiv:2206.11795 [cs].

Steven Bird and Edward Loper. 2004. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Julia Bonn, Martha Palmer, Zheng Cai, and Kristin Wright-Bettner. 2020. Spatial AMR: Expanded Spatial Annotation in the Context of a Grounded Minecraft Corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4883–4892, Marseille, France. European Language Resources Association.

Kaylee Burns, Christopher D. Manning, and Li Fei-Fei. 2021. Neural Abstructions: Abstractions that Support Construction for Grounded Language Learning. ArXiv:2107.09285 [cs].

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10.

Yuhong Deng, Di Guo, Xiaofeng Guo, Naifu Zhang, Huaping Liu, and Fuchun Sun. 2021. MQA: Answering the Question via Robotic Manipulation. In *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation.

Vishnu Sashank Dorbala, Prasoon Goyal, Robinson Piramuthu, Michael Johnston, Dinesh Manocha, and Reza Ghanadhan. 2024. S-EQA: Tackling Situational Queries in Embodied Question Answering. ArXiv:2405.04732 [cs].

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. 2022. MineDojo: Building Open-Ended Embodied Agents with Internet-Scale Knowledge. ArXiv:2206.08853 [cs] version: 2.

Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. 2018. IQA: Visual Question Answering in Interactive Environments. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4089–4098, Salt Lake City, UT. IEEE.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 6904–6913.

William H. Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. 2019. MineRL: A Large-Scale Dataset of Minecraft Demonstrations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 2442–2448, Macao, China. International Joint Conferences on Artificial Intelligence Organization.

Micah Hodosh and Julia Hockenmaier. 2016. Focused Evaluation for Image Description with Binary Forced-Choice Tasks. In *Proceedings of the 5th Workshop on Vision and Language*, pages 19–28, Berlin, Germany. Association for Computational Linguistics.

Nikolai Ilinykh, Yasmeen Emampoor, and Simon Dobnik. 2022. Look and Answer the Question: On the Role of Vision in Embodied Question Answering. In *Proceedings of the 15th International Conference on Natural Language Generation*, pages 236–245, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.

Prashant Jayannavar, Anjali Narayan-Chen, and Julia Hockenmaier. 2020. Learning to execute instructions in a Minecraft dialogue. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2589–2602, Online. Association for Computational Linguistics.

Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. 2016. The Malmo Platform for Artificial Intelligence Experimentation. In *Proceedings of*

the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI'16, pages 4246–4247. AAAI Press. Place: New York, New York, USA.

Julia Kiseleva, Ziming Li, Mohammad Aliannejadi, Shrestha Mohanty, Maartje ter Hoeve, Mikhail Burtsev, Alexey Skrynnik, Artem Zholus, Aleksandr Panov, Kavya Srinet, Arthur Szlam, Yuxuan Sun, Katja Hofmann, Michel Galley, and Ahmed Awadallah. 2021. NeurIPS 2021 Competition IGLU: Interactive Grounded Language Understanding in a Collaborative Environment. ArXiv:2110.06536 [cs].

Nikita Kitaev and Dan Klein. 2017. Where is Misty? Interpreting Spatial Descriptors by Modeling Regions in Space. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 157–166, Copenhagen, Denmark. Association for Computational Linguistics.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Sasha Sax, and Aravind Rajeswaran. 2024. OpenEQA: Embodied Question Answering in the Era of Foundation Models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative Dialogue in Minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. ArXiv:2212.04356 [cs, eess].

Allen Z. Ren, Jaden Clark, Anushri Dixit, Masha Itkina, Anirudha Majumdar, and Dorsa Sadigh. 2024. Explore until Confident: Efficient Exploration for Embodied Question Answering. ArXiv:2403.15941 [cs].

Zhengxiang Shi, Yue Feng, and Aldo Lipani. 2022. Learning to Execute Actions or Ask Clarification Questions. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2060–2070, Seattle, United States. Association for Computational Linguistics.

Kavya Srinet, Yacine Jernite, Jonathan Gray, and Arthur Szlam. 2020. CraftAssist Instruction Parsing: Semantic Parsing for a Voxel-World Assistant. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4693–4714, Online. Association for Computational Linguistics.

Arthur Szlam, Jonathan Gray, Kavya Srinet, Yacine Jernite, Armand Joulin, Gabriel Synnaeve, Douwe Kiela, Haonan Yu, Zhuoyuan Chen, Siddharth Goyal, Demi Guo, Danielle Rothermel, C. Lawrence Zitnick, and Jason Weston. 2019. Why Build an Assistant in Minecraft? ArXiv:1907.09273 [cs].

Sinan Tan, Mengmeng Ge, Di Guo, Huaping Liu, and Fuchun Sun. 2023. Knowledge-Based Embodied Question Answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):11948–11960.

Jesse Thomason, Daniel Gordon, and Yonatan Bisk. 2019. Shifting the Baseline: Single Modality Performance on Visual Navigation & QA. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1977–1983, Minneapolis, Minnesota. Association for Computational Linguistics.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An Open-Ended Embodied Agent with Large Language Models. ArXiv:2305.16291 [cs].

Licheng Yu, Xinlei Chen, Georgia Gkioxari, Mohit Bansal, Tamara L. Berg, and Dhruv Batra. 2019. Multi-Target Embodied Question Answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6302–6311, Long Beach, CA, USA. IEEE.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. 2016. Modeling Context in Referring Expressions. In *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 69–85, Cham. Springer International Publishing.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and Yang: Balancing and Answering Binary Visual Questions. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5014–5022, Las Vegas, NV, USA. IEEE.

A Question Collection

We recruited[5] twelve pairs of self-identified experienced Minecraft players on a college campus in the United States to play mini-games in Minecraft. Participants were at least 18 years old and were compensated $15 in gift cards for a one-hour session, with the opportunity to win an additional $50 gift card for high scores. The prize incentivized

[5]All procedures were reviewed and approved according to the policies of the IRB at the university where the research was conducted. Informed consent was obtained from all participants. Recruitment materials included posters, emails, and an in-class announcement. Templates of consent forms, recruitment materials, and full instructions to the participants are available by request to the corresponding author.

Q:	What do you see?
A:	Grass fields, lots of trees, and flowers.
Q:	What's the flowers or colors?
A:	Red ones, yellow ones, pinkish ones, yellow and white ones.
	...
Q:	Do you see any animals?
A:	I see... Sheep...

Table 4: Dialog from a session of the can-you-do-it game where the questioner is trying to find out if it is possible to dye a sheep orange given what's available in the environment.

teams to work together to achieve their assigned tasks.

Player roles were questioner and agent. Games were designed so that a questioner needs to obtain information about an environment that only the agent can observe. The questioner could ask questions about the agent's environment but could not directly observe it. The agent could move freely around the environment and answer questions but not volunteer information.

(a) Initial screenshot from agent's environment

(b) Screenshot when agent tries to answer "Do you see any animals?"

Figure 2: Screenshots from the can-you-do-it game. The questioner knows the task is "dye a sheep orange," but the agent does not.

build	create	get	make	open	smelt
cook	destroy	grow	mine	place	trade
craft	dig	kill	obtain	put	

Table 5: Teams lost points for using these words that encouraged modification of the environment.

Mini-game 1: In the *can-you-do-it* game, the questioner is given a secret goal task and must determine whether it can be accomplished in the provided environment. To increase the variety of questions asked, we follow Yu et al. (2016) in providing a set of "taboo" words that the questioner may not say unless the agent says them first. For example, Table 4 and Figure 2 show dialog and screenshots from a team trying to determine whether the task "dye a sheep orange" can be completed in the given environment. Taboo words were *yellow*, *red*, *orange*, *sheep*, and *wool*. As an experienced Minecraft player, the questioner knew that orange dye could be made from red and yellow flowers. The agent does not know what the task is or what the taboo words are. The questioner needs the agent's help to gather information about the environment in order to correctly determine whether the materials necessary to dye a sheep orange can be obtained. The team gains points for correctly determining whether a task is or is not feasible and loses points for an incorrect answer. Points are also deducted if the questioner says any taboo words before the agent says them.

In our internal testing, we found that the Minecraft game mechanics make modifying the environment a valuable way of gathering information. However, models for active question answering should ideally limit the changes they make to their environment; a robot butler should not destroy the kitchen wall to determine how much milk is in the refrigerator. The game therefore applied a point penalty for modifying the environment. Points are also deducted if the questioner uses "doing words" from the set listed in Table 5; this discourages the questioner from disguising instructions as questions (e.g., asking "Can you mine three stone?" to instruct the agent to collect stone).

We recorded and transcribed spoken dialog for each session. Transcription was done in two stages: one automated pass using Whisper[6] (Radford et al., 2022), followed by manual corrections. We used

[6] https://github.com/openai/whisper, version 20231117

NLTK v.3.8.1 (Bird and Loper, 2004) to split the transcript into sentences and treated any sentence with a question mark as a question.

Mini-game 2: In the *spot-the-difference* game, the questioner and agent are placed in variations of the same environment. They earn points by identifying true differences and lose points for false positives (Figure 3). Due to the difficulties with transcription in the can-you-do-it game, we shifted to using a custom chat interface (Figure 4). The questioner's interface would only send messages that included at least four words and included a question mark. The agent's interface would only send messages that included at most ten words. Both interfaces required correct spelling before they would send a message. Both players could only see one question and answer at a time, in an effort to reduce the number of context-dependent questions. An admin could see the entire dialog and the questioner's guesses for how the two environments differed.

B Question Review

Data from both games required review to identify usable questions. A question is suitable for EQA only if the correct answer to the question depends on the environment; thus, general knowledge questions such as "What is a baby tree called?" are not EQA questions. In the current work, we ensure that each question can stand alone without dialog context; EQA in dialog is a more difficult problem that we leave for future work. We therefore annotated each question as (1) suitable for EQA, (2) suitable for EQA if context is provided, or (3) unsuitable for EQA. At least two annotators categorized each question. Where we found disagreements, an additional annotator broke the tie. Category 1 questions went directly into the dataset. Category 3 questions were discarded. One annotator reviewed the five previous dialog turns for all category 2 questions and rephrased them to contain context where possible (e.g., turning "What colors are they?" into "What colors are the flowers?"); the rephrased questions became part of the dataset. Questions that could not be rephrased based on the previous five turns were discarded.

C Example Data

Table 6 provides ten randomly selected multiple choice questions from PQB-EQA.

D GPT Settings

We used gpt-4o-2024-08-06, the default GPT4-o model as of this writing, with temperature set to 0.2 and max_tokens at 10 for all experiments.

For the blind condition, the prompt for system role was *Your goal is to answer multiple choice questions about a Minecraft environment. Each question has two possible answers, A and B. Your response should be a letter, either A or B. If you cannot determine the correct answer, guess.*

For the vision condition, the prompt for system role was *Your goal is to answer multiple choice questions based on Minecraft screenshots. Each image is a sequence of screenshots from a Minecraft session. Each question has two possible answers, A and B. Your response should be a letter, either A or B. If you cannot determine the correct answer, guess.*

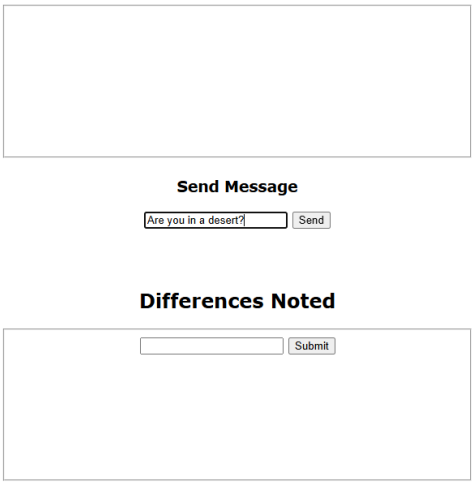
Both models returned text that began with either A or B for every question. Some included additional text after the letter. For the first 350 questions, we confirmed that the trailing text was the answer choice associated with that letter or the first several words of it. We therefore interpret the letter as the model's answer and discard any trailing text.

The cost for all experiments using GPT-4o models totaled $1.49 U.S.

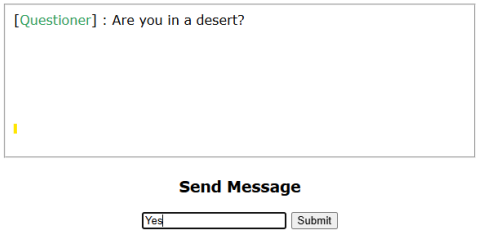
(a) Screenshot from questioner's environment

(b) Screenshot from agent's environment

Figure 3: Example of two environments from the spot-the-difference game. The team would earn points for noting that one environment includes a hay wagon and the other does not or that only one environment has cobwebs on the buildings, but would lose them if they said the environments were different biomes.



(a) Questioner chat interface for asking questions and noting differences



(b) Agent chat interface for answering questions

Figure 4: Chat interface for the spot-the-difference game.

what type of fence is around the garden? A. stone brick wall B. cobblestone wall
Can you access stone? A. yes B. no
when you are on the top of the higher side, how many cacti are there? A. 18 cacti B. 17 cacti
Okay, are there any mobs passive or non-passive in your environment? A. yes B. no
And there are no chests? A. incorrect B. correct
what types of trees are around you? A. there are warped trees B. there are none
do you have redstone? A. yes B. no
what does your environment look like, are there any structures? A. i see ice, blue ice, spruce leaves, stone, snow i also see a(n) house and a(n) mountain, and 4 rabbits, and 1 polar bear B. i see cut sandstone, sandstone, cactus, sand, smooth sandstone i also see a(n) house and a(n) cactus, and 5 rabbits
You don't see any stone at all in any chests or in the ground? A. incorrect B. correct
Is there any vegetation? A. there are warped trees, and crimson trees B. there are none

Table 6: Example questions and answers from PQB-EQA

Learning Sparsity for Effective and Efficient Music Performance Question Answering

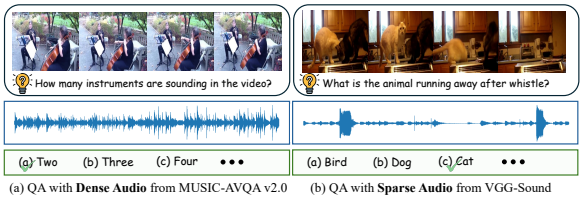
Xingjian Diao[1], **Tianzhen Yang**[2], **Chunhui Zhang**[1],
Weiyi Wu[1], **Ming Cheng**[1], **Jiang Gui**[1]
[1]Dartmouth College, [2]Yale University
xingjian.diao.gr@dartmouth.edu

Abstract

Music performances, characterized by dense and continuous audio as well as seamless audio-visual integration, present unique challenges for multimodal scene understanding and reasoning. Recent Music Performance Audio-Visual Question Answering (Music AVQA) datasets have been proposed to reflect these challenges, highlighting the continued need for more effective integration of audio-visual representations in complex question answering. However, existing Music AVQA methods often rely on dense and unoptimized representations, leading to inefficiencies in the isolation of key information, the reduction of redundancy, and the prioritization of critical samples. To address these challenges, we introduce `Sparsify`, a sparse learning framework specifically designed for Music AVQA. It integrates three sparsification strategies into an end-to-end pipeline and achieves state-of-the-art performance on the Music AVQA datasets. In addition, it reduces training time by 28.32% compared to its fully trained dense counterpart while maintaining accuracy, demonstrating clear efficiency gains. To further improve data efficiency, we propose a key-subset selection algorithm that selects and uses approximately 25% of MUSIC-AVQA v2.0 training data and retains 70–80% of full-data performance across models.

1 Introduction

Music performances, with their dense, continuous audio and seamless audio-visual integration, present both challenges and opportunities for multimodal scene understanding and reasoning (You et al., 2025; Diao et al., 2024). To address the complexities of audio-visual reasoning in music scenarios, the task of Music Performance Audio-Visual Question Answering (Music AVQA) has been proposed, and the corresponding datasets, MUSIC-AVQA (Li et al., 2022) and its extended version MUSIC-AVQA v2.0 (Liu et al., 2024), with an example shown in Figure 1(a), have been introduced to facilitate research in this emerging area.



(a) QA with **Dense Audio** from MUSIC-AVQA v2.0 (b) QA with **Sparse Audio** from VGG-Sound

Figure 1: Dense Audio QA (Liu et al., 2024) vs. Sparse Audio QA (Chen et al., 2020). Music performances contain dense and continuous audio signals with substantial inherent redundancy, much of which is irrelevant to the question being asked. **Sparse learning** has the potential to effectively filter out such redundancies, enabling more efficient and accurate reasoning.

Existing AVQA methods for music performances have evolved from early cross-modality learning approaches developed for speech recognition (Ngiam et al., 2011; Srivastava and Salakhutdinov, 2012) to more recent advancements in multimodal fusion (Yun et al., 2021; Yang et al., 2022), positive-negative pair construction (Li et al., 2022), and state-of-the-art models such as LAVisH (Lin et al., 2023), which adapts pretrained ViTs for cross-modal learning, and DG-SCT (Duan et al., 2023), which employs audio-visual prompts within frozen encoders to enhance reasoning. However, current Music AVQA methods face significant limitations in effectively modeling sparse representations, which are crucial for addressing the unique challenges posed by Music AVQA tasks. These limitations include: ① an overreliance on dense, unoptimized representations that struggle to isolate key information from dense audio-visual signals (Ye et al., 2024; Diao et al., 2024); ② a lack of effective redundancy reduction mechanisms, resulting in inefficiencies during feature extraction and model inference (Shang et al., 2024); ③ the absence of prioritization strategies to identify task-critical samples, which limits scalability and prolongs training times (Qin et al., 2024; Li et al., 2023a).

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 136–146
July 27 - August 1, 2025

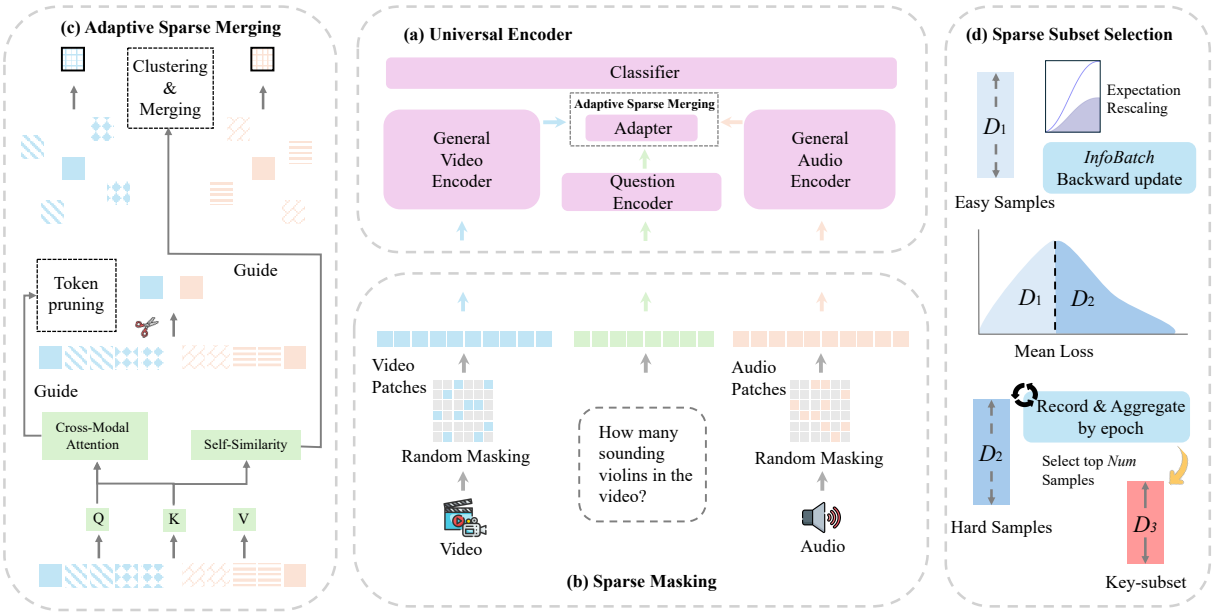


Figure 2: Overview of the `Sparsify` framework. `Sparsify` integrates a (a) Universal Encoder and three sparsification components: (b) Sparse Masking to reduce redundancy by masking audio and visual tokens; (c) Adaptive Sparse Merging to select and merge key multimodal tokens based on similarity; and (d) Sparse Subset Selection to prioritize impactful samples and reweight gradients with $InfoBatch$ (Qin et al., 2024).

To address these limitations, we propose `Sparsify`, a sparse learning framework designed for Music AVQA tasks. Our contributions are:

- We present an end-to-end pipeline for Music AVQA that integrates three sparsification strategies and demonstrate its effectiveness with state-of-the-art results on the MUSIC-AVQA datasets.
- `Sparsify` reduces training time by 28.32% while maintaining the accuracy of question answering compared to fully trained dense models, demonstrating notable efficiency improvements.
- We introduce a key-subset selection algorithm that reduces the training dataset size by approximately 75%, while retaining about 70-80% of the original performance across AVQA models.

2 `Sparsify` Framework

2.1 Learning Multimodal Representations

`Sparsify` adapts the Amuse framework (Diao et al., 2024) as its Universal Encoder, which includes a General Video Encoder built on Swin-V2 (Liu et al., 2022), a General Audio Encoder leveraging the HTS-Audio Transformer (Chen et al., 2022a), and a Question Encoder based on a standard language transformer (Vaswani et al., 2017), as shown in Figure 2(a). Cross-modal attention is applied to align features across modalities, followed by activation and linear transformation layers, resulting in unified and informative multimodal representations tailored to Music AVQA tasks.

2.2 Sparse Masking for Redundancy Reduction

Music performance data inherently contain substantial redundancies, which pose significant challenges to efficient multimodal learning. Sparse Masking addresses this issue by enforcing structured sparsity to reduce redundancy and enhance computational efficiency. As illustrated in Figure 2(b), the method draws inspiration from recent advances in random masking for multimodal models (Li et al., 2023a). This approach aligns with the objectives of sparse learning, aiming to improve efficiency while preserving model performance.

In the visual modality, Sparse Masking is applied to randomly mask 50% of the image patches, reducing input redundancy and introducing structured sparsity to encourage more efficient visual encoding. For the audio modality, we first convert raw waveforms into mel-spectrograms and apply the same masking ratio to ensure comparable sparsity levels. This unified masking design supports coherent sparsity across modalities through consistent information reduction, contributing to more effective multimodal representation learning.

2.3 Sparse Merging for Token Consolidation

In music performance AVQA tasks, dense multimodal inputs often include redundant tokens that unnecessarily increase computational overhead while diluting critical task-relevant information. To tackle this challenge, we adapt the PruMerge method proposed by Shang et al. (2024), applying it to the audio-visual setting as illustrated in Figure 2(c). This strategy dynamically prioritizes and consolidates tokens based on their significance, aligning with the objectives of sparse learning to enhance efficiency and preserve meaningful representations. Following Shang et al. (2024), our approach evaluates token importance using cross-modal attention scores $\mathbf{a} = \text{softmax}\left(\frac{\mathbf{Q}\cdot\mathbf{K}^{\mathbf{T}}}{\sqrt{d}}\right)\mathbf{V}$, where query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) interactions highlight the relevance of each token. We then apply the Interquartile Range (IQR) method to these scores, dynamically identifying tokens within the top quartile of importance. IQR is particularly effective for filtering noise and ensuring robustness in token prioritization by focusing on outliers that represent highly salient features. Once the key tokens are identified, remaining tokens are clustered and adaptively merged with the closest key tokens according to their similarity, calculated as $\text{Similarity}(\mathbf{tok}_i, \mathbf{tok}_j) = \mathbf{k}_i \cdot \mathbf{k}_j^T$. This merging process retains critical tokens while integrating complementary features, preserving representational integrity. Sparse Merging ensures efficient multimodal integration with aligned sparsity across audio and visual modalities.

2.4 Sparse Subset Selection for Efficient Training

Training on dense audio-visual datasets is computationally expensive due to excessive redundancy. Sparse Subset Selection, illustrated in Figure 2 (d), addresses this by identifying and focusing on a key subset of samples that contribute the most to learning, significantly reducing training costs while preserving performance.

Following Qin et al. (2024), our method divides samples into "hard-to-learn" (D_1) and "easy-to-learn" (D_2) categories based on their loss values relative to the mean. Hard samples (D_1) are recorded and aggregated by epoch, with their importance weighted by a decay ratio r over k-epoch intervals. This ensures that difficult samples are prioritized early in training, while less critical samples are deprioritized over time. The top num samples with the highest aggregated scores are selected to form the final Key-subset (D_3). $InfoBatch$ (Qin et al., 2024) is used to rescale gradients, pruning redundant "easy-to-learn" samples (D_2) and ensuring that the reduced dataset retains the statistical properties of the original. This combination minimizes redundancy, accelerates convergence, and maintains task performance. The detailed Key-subset Selection Algorithm is presented as Algorithm 1.

Algorithm 1: Key-subset Selection Algorithm

Input: Model M, Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ with N samples, Loss function L, Number of epochs E, Merge group size k, Decrement ratio r, Number of key samples n
Output: Key-subset indices $\mathcal{K}$

```
# Initialization
Initialize scores vector s ← 1 ∈ ℝ^N
Initialize epochs list EpochsList ← [ ]
# Compute original losses
for i ← 1 to N do
    Compute loss l_i ← L(M(x_i), y_i)
    Update score s_i ← l_i
# InfoBatch
for epoch e ← 1 to E do
    Initialize temporary count vector t ← 0 ∈ ℝ^N
    Compute mean loss μ ← (1/N) Σ_{i=1}^N s_i
    for i ← 1 to N do
        Compute loss l_i ← L(M(x_i), y_i)
        Update score s_i ← l_i
        if s_i > μ then
            Increment count t_i ← t_i + 1
    Append t to EpochsList
# Merge
Initialize merged scores m ← 0 ∈ ℝ^N
Compute number of groups G ← ⌈E/k⌉
for group g ← 1 to G do
    Compute group weight w_g ← r^(g−1)
    for epoch e ← (g − 1) · k + 1 to min(g · k, E)
    do
        Update merged scores
        m ← m + w_g · EpochsList[e]
# Select the top n indices as the Key-subset
K ← argsort(−m)[: n]
return Key-subset indices K
```

3 Experiments

3.1 Setup

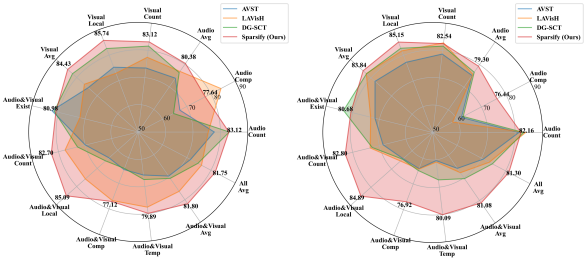
Music AVQA Datasets *(i)* MUSIC-AVQA (Li et al., 2022): Contains 9,288 videos (150 hours) spanning 22 instruments, with 45,867 QA pairs derived from 33 templates across four categories: String, Wind, Percussion, and Keyboard. Each video includes approximately five QA pairs on average. *(ii)* MUSIC-AVQA v2.0 (Liu et al., 2024): An extension addressing data bias, introducing 1,230 additional videos and 8,100 new QA pairs.

Method	Audio-related QA			Visual-related QA			Audio&Visual-related QA						Avg
	Count	Comp	Avg	Count	Local	Avg	Exist	Count	Local	Comp	Temp	Avg	
MUSIC-AVQA													
AVST (Li et al., 2022)	77.78	67.17	73.87	73.52	75.27	74.40	**82.49**	69.88	64.24	64.67	65.82	69.53	71.59
LAVisH (Lin et al., 2023)	75.59	**84.13**	76.86	77.45	72.91	76.29	71.91	77.52	75.81	76.75	77.62	76.31	76.10
DG-SCT (Duan et al., 2023)	**83.27**	64.56	76.34	81.57	82.57	82.08	81.61	72.84	65.91	64.22	67.48	70.56	74.62
Sparsify (Ours)	83.12	77.64	**80.38**	**83.12**	**85.74**	**84.43**	80.98	**82.70**	**85.09**	**77.12**	**79.89**	**81.80**	**81.75**
MUSIC-AVQA v2.0													
AVST (Li et al., 2022)	81.38	61.82	75.20	78.72	77.29	78.05	71.63	68.62	64.39	64.03	60.29	65.83	70.83
LAVisH (Lin et al., 2023)	**83.82**	58.19	75.72	**82.81**	81.73	82.30	73.26	73.45	65.64	64.26	60.82	67.75	73.28
DG-SCT (Duan et al., 2023)	83.13	62.54	76.62	81.61	82.76	82.19	**83.43**	72.70	64.65	64.78	67.34	70.38	74.53
Sparsify (Ours)	82.16	**76.44**	**79.30**	82.54	**85.15**	**83.84**	80.68	**82.80**	**84.89**	**76.92**	**80.09**	**81.08**	**81.30**

Table 1: Comparison with state-of-the-art methods on the MUSIC-AVQA and MUSIC-AVQA v2.0 test set. Accuracy is reported for Audio (Counting, Comparative), Visual (Counting, Location), and Audio-Visual (Existential, Counting, Location, Comparative, Temporal) question types. Average accuracies for Audio, Visual, Audio-Visual, and Overall are also included. **Bold** marks the best results, and underlined marks the second-best.

Full Dataset Training Configuration For the experiments described in Section 3.2, Sparse Masking is applied during the first three epochs and is disabled thereafter. Adaptive Sparse Merging and $InfoBatch$ are used throughout the training. We set the masking rate to 50% for Sparse Masking. For $InfoBatch$, the ratio is set to 0.5 and the delta to 0.875, following the setup in (Qin et al., 2024).

Key-subset Selection Configuration In the key-subset selection, we apply a one-epoch warm-up phase followed by 15 epochs of training. The hyperparameters are set as follows: $N = 15$, $k = 3$, $r = 0.618$, and $Num = 10{,}819$ (i.e., the number of QA pairs). During this stage, only $InfoBatch$ (Qin et al., 2024) is employed, while Sparse Masking and Adaptive Sparse Merging are disabled.



(a) Results on MUSIC-AVQA (b) Results on MUSIC-AVQA v2.0

Figure 3: Radar charts comparing Sparsify with state-of-the-art methods on MUSIC-AVQA and MUSIC-AVQA v2.0, across various question types.

3.2 Comparison with State-of-the-Art

Table 1 presents a detailed comparison between Sparsify and three strong baselines—AVST (Li et al., 2022), LAVisH (Lin et al., 2023), and DG-SCT (Duan et al., 2023)—on both the MUSIC-AVQA and MUSIC-AVQA v2.0 benchmarks. Sparsify achieves the highest overall average accuracy on both datasets, with consistent gains across audio, visual, and audio-visual question types. These improvements highlight the potential of sparse learning in handling the dense and continuous nature of music performance videos.

Sparse learning benefits visual question answering by promoting compact visual representations. On visual-related QA, Sparsify achieves accuracy scores of 84.43% and 83.84% on MUSIC-AVQA and MUSIC-AVQA v2.0, respectively, outperforming DG-SCT by +2.35% and +1.65%, and surpassing LAVisH by +8.14% and +1.54%. These improvements reflect the advantage of sparse inputs in retaining essential spatial and structural cues while reducing visual redundancy. In music performance QA, such representations better support reasoning over complex scenes involving performer locations, interactions, and visual composition. By limiting the influence of background clutter and redundant details, sparse visual representation enables the model to perform more robustly across diverse and fine-grained visual reasoning types.

Sparse learning supports audio question answering by reducing spectral redundancy and enabling efficient acoustic encoding. On audio-related QA, Sparsify achieves gains of +3.52% and +3.58% over LAVisH on MUSIC-AVQA and MUSIC-AVQA v2.0, respectively, and outperforms DG-SCT by +4.04% and +2.68%. These improvements suggest that sparse input representations help suppress redundant frequency patterns while retaining sufficient acoustic detail for question-relevant reasoning. By pruning less informative segments in the spectrogram, Sparsify yields more compact yet informative representations, supporting improved performance on audio-based queries involving complex musical content.

Sparse learning improves audio-visual ques-

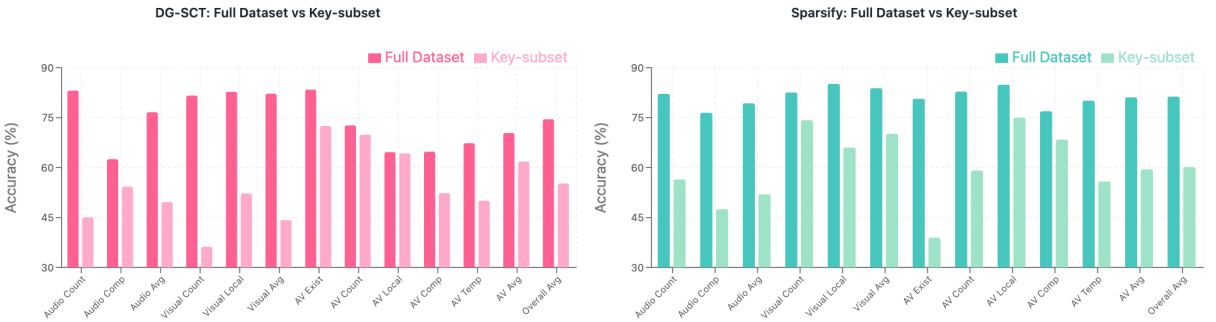


Figure 4: Accuracy comparison of DG-SCT and `Sparsify` trained on the full dataset and the key-subset (∼25% of data) (Liu et al., 2024). Training on the key-subset maintains strong performance despite substantial data reduction.

tion answering by jointly reducing modality-specific redundancies. On audio-visual QA, `Sparsify` outperforms DG-SCT by +11.24% and +10.70%, and LAVisH by +5.49% and +13.33% on MUSIC-AVQA and MUSIC-AVQA v2.0, respectively. These consistent gains suggest that sparsification across both audio and visual modalities helps eliminate less informative content and produce more streamlined multimodal representations with reduced noise. This joint reduction in redundancy allows the model to more effectively capture relevant cross-modal associations in complex performance scenarios.

3.3 Key-subset Selection and Performance Retention

We evaluate the effectiveness of our Key-subset selection algorithm on the MUSIC-AVQA v2.0 dataset (Liu et al., 2024), as illustrated in Figure 4. The selected subset comprises only ∼25% of the original training set (10,819 samples), yet enables models to retain a substantial portion of their full-data performance. When trained exclusively on this subset, `Sparsify` achieves 60.17% accuracy and DG-SCT (Duan et al., 2023) achieves 55.21%, corresponding to 74.01% and 74.08% of their respective accuracies when trained on the full dataset. These results demonstrate that our Key-subset Selection reduces training data usage while retaining much of the models' original performance, offering a data-efficient solution for Music AVQA.

3.4 Training Efficiency Gains from Sparse Learning

Figure 5 illustrates the training efficiency gains of `Sparsify`, which reduces total training time from 173 hours to 124 hours—a 28.32% improvement over its dense variant. These gains reflect the combined effect of three sparsification strategies integrated into the training pipeline. Sparse Masking reduces early-stage computational load by masking 50% of audio and visual tokens. Sparse Merging compresses intermediate representations by consolidating similar tokens, reducing token-level complexity. In parallel, using $InfoBatch$ enhances efficiency by emphasizing harder-to-learn samples, which accelerates convergence and reduces the number of required optimization steps.

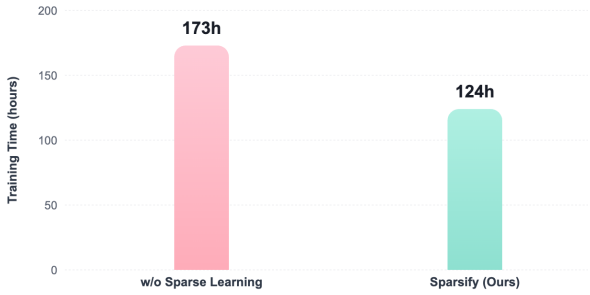


Figure 5: Comparison of the training time of `Sparsify` with a dense variant that disables all three sparsification strategies. Results are reported on the MUSIC-AVQA v2.0 dataset (Liu et al., 2024).

4 Conclusion

We present `Sparsify`, a sparse learning framework for Music AVQA that addresses the inefficiencies inherent in dense audio-visual representations. `Sparsify` achieves this by *(i)* integrating three sparsification strategies into an end-to-end pipeline and achieving state-of-the-art performance on Music AVQA datasets; *(ii)* reducing training time by 28.32% while maintaining comparable accuracy to its dense counterpart. In addition, we propose a key-subset selection algorithm that selects and uses approximately 25% of the MUSIC-AVQA v2.0 training data, while retaining 70–80% of full-data performance across models. We hope our work offers insights into efficient multimodal understanding in dense audio-visual settings.

Limitations

The effectiveness of `Sparsify` has been demonstrated on large-scale Music AVQA benchmarks, as it is specifically designed to address the inefficiencies of dense audio-visual representations in this domain. While it may hold promise for broader multimodal tasks, its behavior in such settings remains to be explored. Extending `Sparsify` to more diverse or unconstrained applications represents a valuable direction for future work.

Ethical Considerations

We examined the study describing the publicly available datasets used in this research and identified no ethical issues regarding the datasets.

Acknowledgment

This study is supported by the Department of Defense grant HT9425-23-1-0267.

References

Asma Ben Abacha and Pierre Zweigenbaum. 2015. Means: A medical question-answering system combining nlp techniques and semantic web technologies. *Information Processing & Management*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *International Conference on Computer Vision*.

Sagar S Arya, Sofia B Dias, Herbert F Jelinek, Leontios J Hadjileontiadis, and Anna-Maria Pappa. 2023. The convergence of traditional and digital biomarkers through ai-assisted biosensing: A new era in translational diagnostics? *Biosensors and Bioelectronics*.

Alexandre Blanco-Gonzalez, Alfonso Cabezon, Alejandro Seco-Gonzalez, Daniel Conde-Torres, Paula Antelo-Riveiro, Angel Pineiro, and Rebeca Garcia-Fandino. 2023. The role of ai in drug discovery: challenges, opportunities, and strategies. *Pharmaceuticals*.

Florian Bordes, Richard Yuanzhe Pang, Anurag Ajay, Alexander C Li, Adrien Bardes, Suzanne Petryk, Oscar Mañas, Zhiqiu Lin, Anas Mahmoud, Bargav Jayaraman, et al. 2024. An introduction to vision-language modeling. *arXiv preprint arXiv:2405.17247*.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. 2023. Audiolm: a language modeling approach to audio generation. *Transactions on Audio, Speech, and Language Processing*.

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. 2020. Vggsound: A large-scale audio-visual dataset. In *International Conference on Acoustics, Speech and Signal Processing*.

Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2022a. Htsat: A hierarchical token-semantic audio transformer for sound classification and detection. In *International Conference on Acoustics, Speech and Signal Processing*.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022b. Convfinqa: Exploring the chain of numerical reasoning in conversational finance question answering. *arXiv preprint arXiv:2210.03849*.

Suresh Dara, Swetha Dhamercherla, Surender Singh Jadav, CH Madhu Babu, and Mohamed Jawed Ahsan. 2022. Machine learning in drug discovery: a review. *Artificial Intelligence Review*.

Soham Deshmukh, Benjamin Elizalde, Rita Singh, and Huaming Wang. 2023. Pengi: An audio language model for audio tasks. *Advances in Neural Information Processing Systems*.

Xingjian Diao, Ming Cheng, and Shitong Cheng. 2023. Av-maskenhancer: Enhancing video representations through audio-visual masked autoencoder. In *International Conference on Tools with Artificial Intelligence*.

Xingjian Diao, Chunhui Zhang, Tingxuan Wu, Ming Cheng, Zhongyu Ouyang, Weiyi Wu, and Jiang Gui. 2024. Learning musical representations for music performance question answering. In *Findings of the Association for Computational Linguistics: EMNLP*.

Xingjian Diao, Chunhui Zhang, Weiyi Wu, Zhongyu Ouyang, Peijun Qing, Ming Cheng, Soroush Vosoughi, and Jiang Gui. 2025. Temporal working memory: Query-guided segment refinement for enhanced multimodal understanding. *arXiv preprint arXiv:2502.06020*.

Haoyi Duan, Yan Xia, Zhou Mingze, Li Tang, Jieming Zhu, and Zhou Zhao. 2023. Cross-modal prompts: Adapting large pre-trained models for audio-visual downstream tasks. In *Advances in Neural Information Processing Systems*.

Haytham M. Fayek and Justin Johnson. 2020. Temporal reasoning via audio question answering. *Transactions on Audio, Speech, and Language Processing*.

Chongyang Gao, Yiren Jian, Natalia Denisenko, Soroush Vosoughi, and VS Subrahmanian. 2024. Gem: generating engaging multimodal content. In *International Joint Conference on Artificial Intelligence*.

Kaixiong Gong, Kaituo Feng, Bohao Li, Yibing Wang, Mofan Cheng, Shijia Yang, Jiaming Han, Benyou

Wang, Yutong Bai, Zhuoran Yang, et al. 2024. Avodyssey bench: Can your multimodal llms really understand audio-visual information? *arXiv preprint arXiv:2412.02611*.

Travis R Goodwin and Sanda M Harabagiu. 2016. Medical question answering for clinical decision support. In *International on Conference on Information and Knowledge Management*.

Yangfan He, Sida Li, Jianhui Wang, Kun Li, Xinyuan Song, Xinhang Yuan, Keqin Li, Kuan Lu, Menghao Huo, Jiaqi Chen, et al. 2025a. Enhancing low-cost video editing with lightweight adaptors and temporal-aware inversion. *arXiv preprint arXiv:2501.04606*.

Yangfan He, Jianhui Wang, Kun Li, Yijin Wang, Li Sun, Jun Yin, Miao Zhang, and Xueqian Wang. 2025b. Enhancing intent understanding for ambiguous prompts through human-machine co-adaptation. *arXiv preprint arXiv:2501.15167*.

Panwen Hu, Nan Xiao, Feifei Li, Yongquan Chen, and Rui Huang. 2023. A reinforcement learning-based automatic video editing method using pre-trained vision-language model. In *International Conference on Multimedia*.

Allen H Huang, Hui Wang, and Yi Yang. 2023. Finbert: A large language model for extracting information from financial text. *Contemporary Accounting Research*.

Vladimir Iashin, Weidi Xie, Esa Rahtu, and Andrew Zisserman. 2022. Sparse in space and time: Audio-visual synchronisation with trainable selectors. *arXiv preprint arXiv:2210.07055*.

Vladimir Iashin, Weidi Xie, Esa Rahtu, and Andrew Zisserman. 2024. Synchformer: Efficient synchronization from sparse cues. In *International Conference on Acoustics, Speech and Signal Processing*.

Yiren Jian, Chongyang Gao, and Soroush Vosoughi. 2023. Bootstrapping vision-language learning with decoupled language pre-training. In *Advances in Neural Information Processing Systems*.

Yiren Jian, Tingkai Liu, Yunzhe Tao, Chunhui Zhang, Soroush Vosoughi, and Hongxia Yang. 2024. Expedited training of visual conditioned language generation via redundancy reduction. In *Annual Meeting of the Association for Computational Linguistics*.

Balaram Yadav Kasula. 2023. Harnessing machine learning for personalized patient care. *Transactions on Latest Trends in Artificial Intelligence*.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. 2023. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in Neural Information Processing Systems*.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. 2018. Tvqa: Localized, compositional video question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Bin Li and Hanjun Deng. 2023. Bilateral personalized dialogue generation with contrastive learning. *Soft Computing*.

Bin Li, Bin Sun, Shutao Li, Encheng Chen, Hongru Liu, Yixuan Weng, Yongping Bai, and Meiling Hu. 2024a. Distinct but correct: generating diversified and entity-revised medical response. *Science China Information Sciences*.

Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. 2019. Controllable text-to-image generation. *Advances in Neural Information Processing Systems*.

Guangyao Li, Yake Wei, Yapeng Tian, Chenliang Xu, Ji-Rong Wen, and Di Hu. 2022. Learning to answer questions in dynamic audio-visual scenarios. In *Conference on Computer Vision and Pattern Recognition*.

Shutao Li, Bin Li, Bin Sun, and Yixuan Weng. 2024b. Towards visual-prompt temporal answer grounding in instructional video. *Transactions on Pattern Analysis and Machine Intelligence*.

Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. 2023a. Scaling language-image pre-training via masking. In *Conference on Computer Vision and Pattern Recognition*.

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023b. Large language models in finance: A survey. In *International Conference on AI in Finance*.

Jinhua Liang, Huan Zhang, Haohe Liu, Yin Cao, Qiuqiang Kong, Xubo Liu, Wenwu Wang, Mark D Plumbley, Huy Phan, and Emmanouil Benetos. 2024. Wavcraft: Audio editing and generation with large language models. *arXiv preprint arXiv:2403.09527*.

Yan-Bo Lin, Yi-Lin Sung, Jie Lei, Mohit Bansal, and Gedas Bertasius. 2023. Vision transformers are parameter-efficient audio-visual learners. In *Conference on Computer Vision and Pattern Recognition*.

Samuel Lipping, Parthasaarathy Sudarsanam, Konstantinos Drossos, and Tuomas Virtanen. 2022. Clotho-aqa: A crowdsourced dataset for audio question answering. In *European Signal Processing Conference*.

Xiulong Liu, Zhikang Dong, and Peng Zhang. 2024. Tackling data bias in music-avqa: Crafting a balanced dataset for unbiased question-answering. In *Winter Conference on Applications of Computer Vision*.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2025. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. 2022. Swin transformer v2: Scaling up capacity and resolution. In *Conference on Computer Vision and Pattern Recognition*.

Dongchan Min, Dong Bok Lee, Eunho Yang, and Sung Ju Hwang. 2021. Meta-stylespeech: Multi-speaker adaptive text-to-speech generation. In *International Conference on Machine Learning*.

Gianluca Monaci, Friedrich T Sommer, and Pierre Vandergheynst. 2008. Learning sparse generative models of audiovisual signals. In *European Signal Processing Conference*.

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *International Conference on Machine Learning*.

Yingwei Pan, Yehao Li, Jianjie Luo, Jun Xu, Ting Yao, and Tao Mei. 2022. Auto-captions on gif: A large-scale video-sentence dataset for vision-language pre-training. In *International Conference on Multimedia*.

Qi Qian, Yuanhong Xu, and Juhua Hu. 2023. Intra-modal proxy learning for zero-shot visual categorization with clip. *Advances in Neural Information Processing Systems*.

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Zhaopan Xu, Daquan Zhou, Lei Shang, Baigui Sun, Xuansong Xie, and Yang You. 2024. Infobatch: Lossless training speed up by unbiased dynamic data pruning. In *International Conference on Learning Representations*.

Khyati Saini and Pardeep Singh. 2023. Evolution of financial question answering themes, challenges, and advances. In *International Conference on Recent Innovations in Computing*.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2024. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*.

Peng Shen, Satoshi Tamura, and Satoru Hayamizu. 2013. Audio-visual interaction in sparse representation features for noise robust audio-visual speech recognition. In *Auditory-Visual Speech Processing*.

Fangxun Shu, Lei Zhang, Hao Jiang, and Cihang Xie. 2023. Audio-visual llm for video understanding. *arXiv preprint arXiv:2312.06720*.

Teotino Gomes Soares, Azhari Azhari, Nur Rokhman, and E Wonarko. 2021. Education question answering systems: a survey. In *International MultiConference of Engineers and Computer Scientists*.

Nitish Srivastava and Russ R. Salakhutdinov. 2012. Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems*.

Tim Steuer, Anna Filighera, and Thomas Tregel. 2022. Investigating educational and noneducational answer selection for educational question generation. *IEEE Access*.

Yi Su, Jisheng Bai, Qisheng Xu, Kele Xu, and Yong Dou. 2025. Audio-language models for audio-centric tasks: A survey. *arXiv preprint arXiv:2501.15177*.

Deeksha Varshney, Aizan Zafar, Niranshu Kumar Behera, and Asif Ekbal. 2023. Knowledge graph assisted end-to-end medical dialog generation. *Artificial Intelligence in Medicine*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Shiru Wang, Yao Chen, Lesley A Jarvis, Yucheng Tang, David J Gladstone, Kimberley S Samkoe, Brian W Pogue, Petr Bruza, and Rongxiao Zhang. 2024. Robust real-time segmentation of bio-morphological features in human cherenkov imaging during radiotherapy via deep learning. *arXiv preprint arXiv:2409.05666*.

Yanbo J Wang, Yuming Li, Hui Qin, Yuhang Guan, and Sheng Chen. 2022. A novel deberta-based model for financial question answering task. *arXiv preprint arXiv:2207.05875*.

Yuancheng Wang, Zeqian Ju, Xu Tan, Lei He, Zhizheng Wu, Jiang Bian, et al. 2023. Audit: Audio editing by following instructions with latent diffusion models. *Advances in Neural Information Processing Systems*.

Yake Wei, Di Hu, Yapeng Tian, and Xuelong Li. 2022. Learning in audio-visual context: A review, analysis, and new perspective. *arXiv preprint arXiv:2208.09579*.

Yuxiang Wei, Anees Abrol, and Vince D Calhoun. 2025. Hierarchical spatio-temporal state-space modeling for fmri analysis. In *International Conference on Research in Computational Molecular Biology*.

Yuxiang Wei, Yuqian Chen, Tengfei Xue, Leo Zekelman, Nikos Makris, Yogesh Rathi, Weidong Cai, Fan Zhang, and Lauren J O'Donnell. 2023. A deep network for explainable prediction of non-imaging phenotypes using anatomical multi-view data. In *International Workshop on Computational Diffusion MRI*.

Haibin Wu, Xuanjun Chen, Yi-Cheng Lin, Kai-wei Chang, Ho-Lam Chung, Alexander H Liu, and Hung-yi Lee. 2024. Towards audio language modeling–an overview. *arXiv preprint arXiv:2402.13236*.

Yongchao Wu, Aron Henriksson, Martin Duneld, and Jalal Nouri. 2023. Towards improving the reliability and transparency of chatgpt for educational question answering. In *European Conference on Technology Enhanced Learning*.

Binzhu Xie, Sicheng Zhang, Zitang Zhou, Bo Li, Yuanhan Zhang, Jack Hessel, Jingkang Yang, and Ziwei Liu. 2024. Funqa: Towards surprising video comprehension. In *European Conference on Computer Vision*.

Pinci Yang, Xin Wang, Xuguang Duan, Hong Chen, Runze Hou, Cong Jin, and Wenwu Zhu. 2022. Avqa: A dataset for audio-visual question answering on videos. In *International Conference on Multimedia*.

Qian Yang, Jin Xu, Wenrui Liu, Yunfei Chu, Ziyue Jiang, Xiaohuan Zhou, Yichong Leng, Yuanjun Lv, Zhou Zhao, Chang Zhou, et al. 2024. Air-bench: Benchmarking large audio-language models via generative comprehension. *arXiv preprint arXiv:2402.07729*.

Jiawei Yao, Qi Qian, and Juhua Hu. 2024a. Customized multiple clustering via multi-modal subspace proxy learning. *arXiv preprint arXiv:2411.03978*.

Jiawei Yao, Qi Qian, and Juhua Hu. 2024b. Multi-modal proxy learning towards personalized visual multiple clustering. In *Conference on Computer Vision and Pattern Recognition*.

Qilang Ye, Zitong Yu, and Xin Liu. 2024. Answering diverse questions via text attached with key audio-visual clues. *arXiv preprint arXiv:2403.06679*.

Wenhao You, Xingjian Diao, Chunhui Zhang, Keyi Kong, Weiyi Wu, Zhongyu Ouyang, Chiyu Ma, Tingxuan Wu, Noah Wei, Zong Ke, Ming Cheng, Soroush Vosoughi, and Jiang Gui. 2025. Music's multimodal complexity in avqa: Why we need more than general multimodal llms. *arXiv preprint arXiv:2505.20638*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Conference on Computer Vision and Pattern Recognition*.

Heeseung Yun, Youngjae Yu, Wonsuk Yang, Kangil Lee, and Gunhee Kim. 2021. Pano-avqa: Grounded audio-visual question answering on 360deg videos. In *International Conference on Computer Vision*.

Chunhui Zhang, Yiren Jian, Zhongyu Ouyang, and Soroush Vosoughi. 2025. Pretrained image-text models are secretly video captioners. *arXiv preprint arXiv:2502.13363*.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024. Vision-language models for vision tasks: A survey. *Transactions on Pattern Analysis and Machine Intelligence*.

Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. 2019. The sound of motions. In *International Conference on Computer Vision*.

Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. 2018. The sound of pixels. In *European Conference on Computer Vision*.

Ziyi Zhou, Ming Cheng, Xingjian Diao, Yanjun Cui, and Xiangling Li. 2024. Glumarker: A novel predictive modeling of glycemic control through digital biomarkers. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. *arXiv preprint arXiv:2105.07624*.

A Baselines

- **AVST** (Li et al., 2022): Integrates audio, visual, and question modalities for spatio-temporal reasoning in audio-visual question answering. It aligns modalities through spatial and temporal grounding, fuses features into a joint representation, and optimizes both grounding and QA objectives.

- **LAVisH** (Lin et al., 2023): Adapts frozen Vision Transformers for audio-visual tasks using lightweight adapters and latent tokens to compress and fuse audio-visual information. Cross-modal attention and adapter modules enable bidirectional interaction between modalities.

- **DG-SCT** (Duan et al., 2023): Enhances audio-visual tasks through a Dual-Guided Spatial-Channel-Temporal attention mechanism, dynamically adjusting feature extraction and facilitating bidirectional audio-visual guidance with lightweight interaction layers.

B Related Work

Audio-Visual Video Understanding Audio and visual modalities offer complementary cues that, when jointly modeled, support a more comprehensive understanding of the scene (Wei et al., 2022; Diao et al., 2023; Shu et al., 2023; Diao et al., 2025). Early work focused on joint representations for tasks like audio-visual speech recognition (Ngiam et al., 2011) and multimodal deep learning (Srivastava and Salakhutdinov, 2012). Recent methods enhance fusion techniques for sound source localization (Zhao et al., 2018) and audio-driven visual analysis (Zhao et al., 2019). Frameworks such as LAVisH (Lin et al., 2023), which proposed a latent

audio-visual hybrid adapter that adapts pretrained ViTs to audio-visual tasks by injecting a small number of trainable parameters into every layer of a frozen ViT, and DG-SCT (Duan et al., 2023) which incorporates trainable cross-modal interaction layers into pre-trained audio-visual encoders, allowing adaptive extraction of crucial information from the current modality across spatial, channel, and temporal dimensions, while preserving the frozen parameters of large-scale pre-trained models. As for benchmarks, there are MUSIC-AVQA (Li et al., 2022), AVQA (Yang et al., 2022), MUSIC-AVQA v2.0 (Liu et al., 2024) and AV-Odyssey Bench (Gong et al., 2024), which focus on whether model can truly understand audio-visual information. However, existing approaches overlook the unique challenges of music performance datasets, where dense and continuous audio-visual signals lead to significant redundancy. These dense representations hinder efficient processing and dilute task-relevant features, necessitating sparsification strategies to enable efficient reasoning in this domain.

Multimodal Question Answering Multimodal question answering spans Visual QA (VQA) (Antol et al., 2015; Lei et al., 2018; Li et al., 2024b), Audio QA (AQA) (Fayek and Johnson, 2020), and Audio-Visual QA (AVQA) (Li et al., 2022), requiring the integration of modality-specific signals for complex reasoning tasks. For VQA, datasets such as MMMU (Yue et al., 2024) and MMBench (Liu et al., 2025) provide carefully curated benchmarks that evaluate vision-language models across diverse domains. For AQA, notable datasets include Clotho-AQA (Lipping et al., 2022) and AIR-Bench (Yang et al., 2024), which consist of question-answering tasks derived from environmental and event-based audio scenes. AVQA benchmarks such as MUSIC-AVQA (Li et al., 2022), AVQA (Yang et al., 2022), Pano-AVQA (Yun et al., 2021), FunQA (Xie et al., 2024), and MUSIC-AVQA v2.0 (Liu et al., 2024) emphasize spatio-temporal reasoning and multimodal fusion in complex video contexts. Among these, Music AVQA presents distinctive challenges due to its continuous and densely structured audio signals, making it valuable for multimodal reasoning (You et al., 2025).

Sparse Learning in Audio-Visual Signals Sparsity has become increasingly crucial in audio-visual signal processing due to the inherent complexity and redundancy of cross-modal data. Early approaches employ shift-invariant kernels (Monaci et al., 2008) to capture essential patterns while reducing computational overhead. This foundation leads to more sophisticated methods using group sparsity and joint dictionaries (Shen et al., 2013), which are particularly effective in handling noisy and variable signals. Current research focuses on temporal dynamics in audio-visual learning, where audio-visual relationships are often intermittent but contextually meaningful (Iashin et al., 2022). Modern transformer-based architectures with specialized selection mechanisms (Iashin et al., 2024) have shown promise in processing extended sequences efficiently. However, sparsity-based approaches remain underexplored in the context of music performance question answering, where challenges such as overlapping instruments and complex audio-visual interactions demand more efficient representations. Our work aims to bridge this gap with sparsification strategies.

C Positioning Music AVQA Among Multimodal Tasks

To contextualize Music AVQA, it is useful to distinguish it from broader multimodal tasks that also integrate information across modalities. This section contrasts Music AVQA with vision-language modeling, audio-language modeling, and other representative domain-specific question answering to highlight its unique challenges and requirements.

C.1 Vision-Language Modeling

Vision-language modeling aims to enable multimodal systems to interpret visual content—such as images and videos—in conjunction with textual descriptions (Jian et al., 2023; Bordes et al., 2024; Jian et al., 2024; Zhang et al., 2024). It has supported a wide range of applications, including text-to-image generation (Li et al., 2019; Gao et al., 2024; He et al., 2025b), video editing (Hu et al., 2023; He et al., 2025a), video captioning and grounding (Pan et al., 2022; Li et al., 2024b; Zhang et al., 2025), and proxy learning (Qian et al., 2023; Yao et al., 2024a,b). In contrast, Music AVQA requires integrated reasoning over continuous audio-visual streams, where visual understanding must be synchronized with rhythm, motion, and acoustic cues. This setting introduces challenges such as temporal alignment and redundancy reduction, which are not typically emphasized in standard vision-language tasks.

C.2 Audio-Language Modeling

Audio-language modeling (Borsos et al., 2023; Deshmukh et al., 2023; Wu et al., 2024; Su et al., 2025) builds systems that fuse audio features with text for various downstream tasks such as audio question answering (Fayek and Johnson, 2020; Lipping et al., 2022), text-to-speech generation (Min et al., 2021; Le et al., 2023), and audio editing (Wang et al., 2023; Liang et al., 2024). These tasks primarily focus on modeling relationships between acoustic signals and language, often in domains such as speech, environmental sounds, or sound events. Unlike conventional audio-language tasks that focus on modeling acoustic-linguistic associations, Music AVQA incorporates a visual modality that is intricately entangled with the audio stream in music performance contexts. This setting necessitates fine-grained multimodal reasoning, where models must jointly interpret auditory patterns, visual dynamics, and their temporal interplay to answer performance-specific questions.

C.3 Domain-Specific Question Answering

Domain-specific question answering systems are designed to operate within specialized fields by leveraging structured knowledge and domain-specific data. Examples such as educational, financial, and medical QA, as discussed below, entail distinct input modalities, reasoning demands, and representational challenges.

Educational QA Educational Question Answering (Educational QA) systems (Soares et al., 2021; Steuer et al., 2022; Wu et al., 2023) are designed to support learning processes by responding to student queries based on educational materials such as textbooks, lecture notes, and academic articles. The primary goal is to clarify concepts, explain solutions, or guide students through subject matter. In contrast, Music AVQA involves perceptual reasoning over evolving audio-visual input. Instead of extracting explicit concepts from structured curricula, models must interpret expressive cues such as gestural nuance, visual-musical alignment, and acoustic articulation in continuous video streams. This shift demands interpreting perceptual and temporal patterns, which are not typically required in conventional educational QA. The emphasis on fluid multimodal integration further distinguishes Music AVQA as a challenging reasoning setting.

Financial QA Financial Question Answering (Financial QA) (Li et al., 2023b; Huang et al., 2023; Saini and Singh, 2023) focuses on extracting insights and answering questions from a wide range of financial data (Chen et al., 2022b), such as company reports, market data, economic indicators, and financial news. These systems assist analysts, investors, and businesses in making informed decisions by providing quick access to relevant financial information and analysis. While Financial QA can involve data from multiple views (e.g., text, tables, charts) (Zhu et al., 2021), it typically does not involve the continuous, dense audio-visual streams found in music performances. The core task in Financial QA is to identify factual information, understand financial terminology, perform numerical reasoning, and interpret trends from often structured or semi-structured textual and numerical data (Wang et al., 2022). In contrast, Music AVQA centers on the temporal and semantic understanding of performance events, requiring models to interpret how visual gestures correspond to musical outcomes, such as identifying sounding instruments, tracking temporal changes, and linking expressive motion to acoustic effects, rather than extracting or reasoning over structured financial data.

Medical QA AI models are increasingly utilized across medical field, tackling a wide array of applications such as diagnostic assistance through analysis of medical images (e.g., X-rays, MRIs) (Wei et al., 2023; Wang et al., 2024; Wei et al., 2025) and dialogues (Varshney et al., 2023; Li et al., 2024a), drug discovery (Dara et al., 2022; Blanco-Gonzalez et al., 2023), digital biomarkers (Arya et al., 2023; Zhou et al., 2024), and personalized patient care (Kasula, 2023; Li and Deng, 2023). Among these, Medical Question Answering (Medical QA) (Abacha and Zweigenbaum, 2015; Goodwin and Harabagiu, 2016) is a specialized field focused on developing systems that understand and respond to health-related queries. These systems often process information from diverse sources to provide accurate medical information or support clinical decision-making. In contrast, Music AVQA centers on interpreting music-related videos, requiring models to reason over dense, continuous audio and tightly synchronized visual streams. While both involve multimodal and complex reasoning, Music AVQA uniquely demands fine-grained alignment of perceptual cues in expressive performance contexts, such as gesture, rhythm, and phrasing.

Cross-Lingual Transfer of Cultural Knowledge: An Asymmetric Phenomenon

Chen Zhang, Zhiyuan Liao, Yansong Feng*
Wangxuan Institute of Computer Technology, Peking University
{zhangch,fengyansong}@pku.edu.cn
liaozy@stu.pku.edu.cn

Abstract

Despite substantial research efforts evaluating how well large language models (LLMs) handle global cultural diversity, the mechanisms behind their cultural knowledge acquisition, particularly in multilingual settings, remain unclear. We study this question by investigating how cultural knowledge transfers across languages during the language adaptation of LLMs, a process where an LLM is continually pre-trained to learn another language. We introduce an interpretable framework to study this transfer, ensuring training data transparency and controlling transfer effects. Through a study of four non-Anglophonic cultures, we observe bidirectional cultural transfer between English and other high-resource languages, while low-resource languages primarily transfer knowledge to English with limited reverse flow. To explain this asymmetric phenomenon, we propose a frequency-based hypothesis: cultural knowledge appearing more frequently in the pretraining data transfers more easily, which is supported by empirical analysis of the training corpora. We hope our findings could inform future research on knowledge transfer and promote the development of culturally aware models, particularly for low-resource languages.

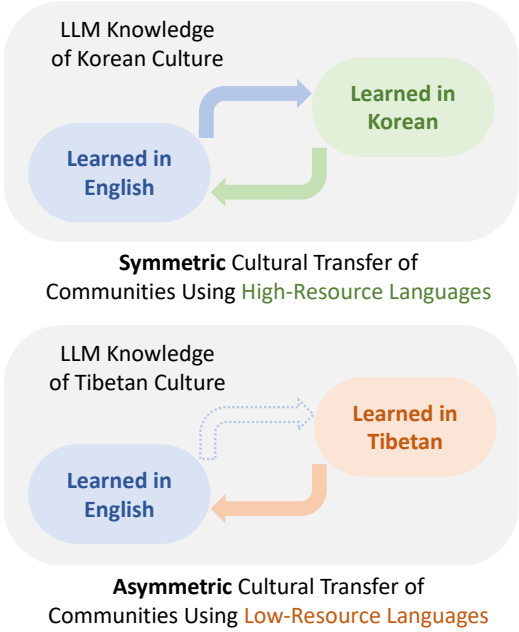


Figure 1: LLMs might exhibit different patterns of cross-lingual transfer of cultural knowledge when continually pretrained in non-English languages. The transfer tends to be bidirectional for cultural knowledge of communities using high-resource languages, but often remains asymmetric for low-resource ones.

1 Introduction

Although large language models (LLMs) have made significant progress in processing diverse languages (Huang et al., 2024; Dang et al., 2024), they face the challenge of addressing the complexity of global cultural diversity (Hershcovich et al., 2022; Pawar et al., 2024; Liu et al., 2024; Pawar et al., 2025). Existing research primarily evaluates whether LLMs possess adequate cultural knowledge of non-English-speaking or non-Anglophonic communities (Yin et al., 2022; Fung et al., 2024; Shi et al., 2024; Li et al., 2024). However, the sources and mechanisms by which LLMs acquire cultural knowledge, particularly in multilingual settings, remain largely unexplored.

In this paper, we investigate the dynamics of cultural knowledge acquisition in LLMs. We particularly focus on the mechanisms of cross-lingual transfer, a widely observed phenomenon when LLMs are adapted to new languages (Ye et al., 2023; Hu et al., 2024; Zhao et al., 2024; Etxaniz et al., 2024). This investigation faces two main challenges. First, the opacity of LLMs' training data and procedures limits the feasibility of conducting interpretable experiments, making it difficult to analyze the sources and influencing factors of knowledge transfer. Second, when performance improvements are observed on cultural knowledge

*Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 147–157
July 27 - August 1, 2025

questions for a given language, it is challenging to disentangle whether these gains result from improved language proficiency or from cross-lingual knowledge transfer.

To address these challenges, we design a new research framework that allows a more controlled and interpretable study on the cross-lingual transfer of cultural knowledge. Our framework has three key features. First, it emphasizes the transparency in training data, making it possible to locate the source of cultural knowledge. Specifically, we pretrain a base model from scratch using carefully filtered English Wikipedia data, followed by continual pretraining on corpora in other languages. Second, our framework isolates the effects of cross-lingual transfer from the improved language proficiency. We employ two continual pretraining settings, one that allows cross-lingual transfer with explicit cross-lingual bridges of parallel co-occurrence and the other that minimizes the chances of such transfer. The performance gap between these two settings serves as an estimate of cross-lingual transfer effects. Third, using bilingual parallel probing questions, our framework can analyze the transfer in both directions: from English to non-English languages and vice versa.

We apply this framework to study the cultural transfer of four non-Anglophonic communities: Koreans in South Korea, Han Chinese, Tibetans, and Mongols in China, whose native languages are Korean (ko), Chinese (zh), Tibetan (bo), and Mongolian (mn), respectively. During continual pretraining on these languages, we observe notable differences in cross-lingual cultural knowledge transfer, as shown in Figure 1. Specifically, for high-resource languages (Chinese and Korean), knowledge transfer occurs bidirectionally between English and the target languages. However, for lower-resource languages (Tibetan and Mongolian), we find an asymmetric transfer pattern: knowledge transfer from low-resource languages to English is more pronounced than the reverse direction.

Based on these observations, we hypothesize that cultural knowledge that appears in the training corpus more frequently is more likely to transfer across languages, inspired by previous findings that frequency of appearance during training is an important factor in the monolingual knowledge acquisition of LLMs (Chang et al., 2024; Li et al., 2025). To test this hypothesis, we analyze our training data by estimating the number of occurrences for each cultural knowledge item in both the English and non-English corpora. Our findings reveal that for low-resource languages, cultural knowledge appears significantly more frequently in non-English corpora than in the English corpus, potentially contributing to the observed asymmetry in transfer. An instance-level analysis further shows that cultural knowledge items that transfer across languages tend to have higher-than-average frequencies of occurrence in the corpus, providing additional evidence for our hypothesis.

Our main contributions are as follows: (1) We introduce a novel framework towards an interpretable study of cross-lingual knowledge transfer, effectively isolating transfer effects from other influencing factors; (2) We investigate the dynamics of knowledge transfer for four non-Anglophonic cultural communities, revealing an asymmetric transfer pattern for low-resource languages. (3) We propose a frequency-based hypothesis to explain the asymmetric transfer phenomenon, supported by empirical analysis over pretraining data. Our data and code are publicly available to the community[1].

2 Methodology

As shown in Figure 2, our framework consists of three steps: pretraining in English, controlled continual pretraining in non-English languages, and bilingual evaluation on cultural questions. It emphasizes transparent training data, decoupling of transfer effects, and bilingual parallel evaluation.

Transparent Pretrainining Instead of using LLMs trained with closed-source training data, we advocate training smaller models from scratch with transparent data. In this way, we can clearly observe the process of knowledge transfer and trace the sources of learned knowledge from the corpus. Specifically, we train a 0.5B model from scratch using an English Wikipedia that filters out all non-Latin characters. Afterwards, we continually pretrain it with corpora in other languages and track how the cultural knowledge transfers throughout the training process.

Decoupling of Transfer Effects in Continual Pretraining To disentangle the effects of cross-lingual transfer from improved language proficiency, we systematically control contributing factors, including language similarities, lexical overlaps, and parallel co-occurrences (Radford et al.,

[1] https://github.com/luciusssss/cross-lingual-culture

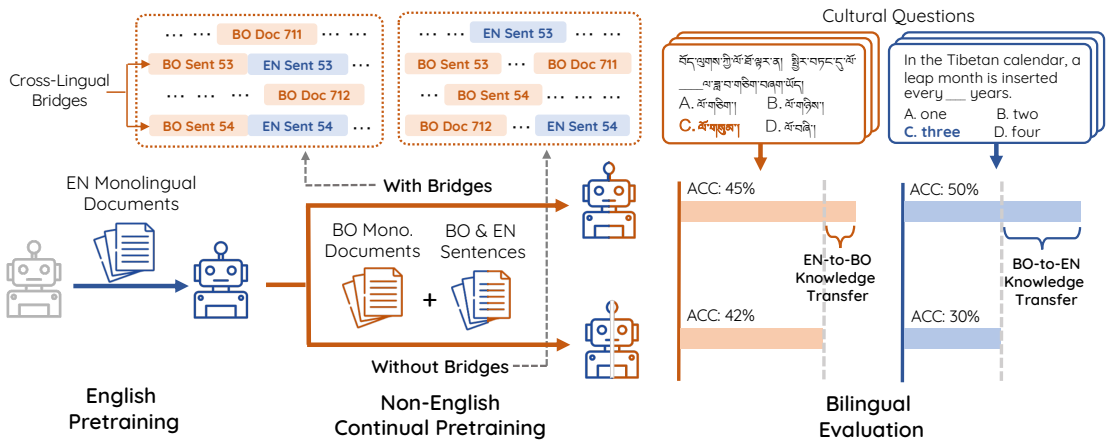


Figure 2: Our framework for studying the cross-lingual transfer of cultural knowledge. We use the transfer between English and Tibetan (bo) as an illustrative example.

2019; Blevins and Zettlemoyer, 2022; Philippy et al., 2023). We design two distinct settings for continual pretraining: one that facilitates cross-lingual transfer and the other that minimizes the chances of transfer as much as possible. The performance gap between these two settings can serve as an indicator of the effect of cross-lingual transfer.

Specifically, we select languages using non-Latin writing systems for study, which naturally have little lexical overlap with English. We further ensure that the English corpus contains no non-Latin characters and the non-English corpora contain no Latin characters, maximizing the isolation between languages during pretraining.

To introduce bridges of cross-lingual transfer to one of the settings, we incorporate parallel sentences into the continual pretraining data, where each pair of parallel sentences is concatenated and mixed with other monolingual training data, as illustrated in Figure 2. This approach has been proven to be effective in helping LLMs learn cross-lingual mappings during pretraining (Anil et al., 2023; Wei et al., 2023; Lin et al., 2024). In contrast, the other setting without cross-lingual bridges uses the same training data, but we purposely prevent paired parallel sentences from co-occurring. Instead, the two sentences in each parallel pair are treated as independent documents, shuffled within the training data. This eliminates the possibility of learning cross-lingual alignment through bilingual co-occurrence in a shared context.

Bilingual Parallel Evaluation For the checkpoints throughout the continual pretraining process, we evaluate them with cultural probing questions in both English and non-English. When using the non-English-version questions for evaluation, the performance gap between the settings with and without bridges indicates the transfer from English to non-English languages. Similarly, the English-version questions are used to probe the transfer from non-English to English.

3 Experiments

Studied Cultures We select four cultural communities, Koreans in South Korea, Han Chinese, Tibetans, and Mongols in China, for our study. Two of them are associated with high-resource languages (Korean and Chinese), and the other two with low-resource languages (Tibetan and Mongolian). See more information about these communities and their cultures in Appendix A. The languages are deliberately chosen to minimize the cross-lingual transfer with English, as discussed in Section 2.

Pretraining We pretrain a model from scratch with the architecture of Qwen-2.5-0.5B (Yang et al., 2024) using English Wikipedia, amounting to 5B tokens. We then continually pretrain it for each non-English language for 1,500 steps. The batch size is set to 0.5M tokens. See details of pretraining corpora and implementation in Appendix C.1.

Evaluation Since small models trained from scratch exhibit limited instruction-following capabilities, we use cloze-style questions for evaluation. For Korean culture, we sample cultural questions from CLIcK (Kim et al., 2024) and convert them into cloze-style format using GPT-4o. For the other three cultures in China, we collect questions from 《中国少数民族》 (*Ethnic Minorities in China*; Na-

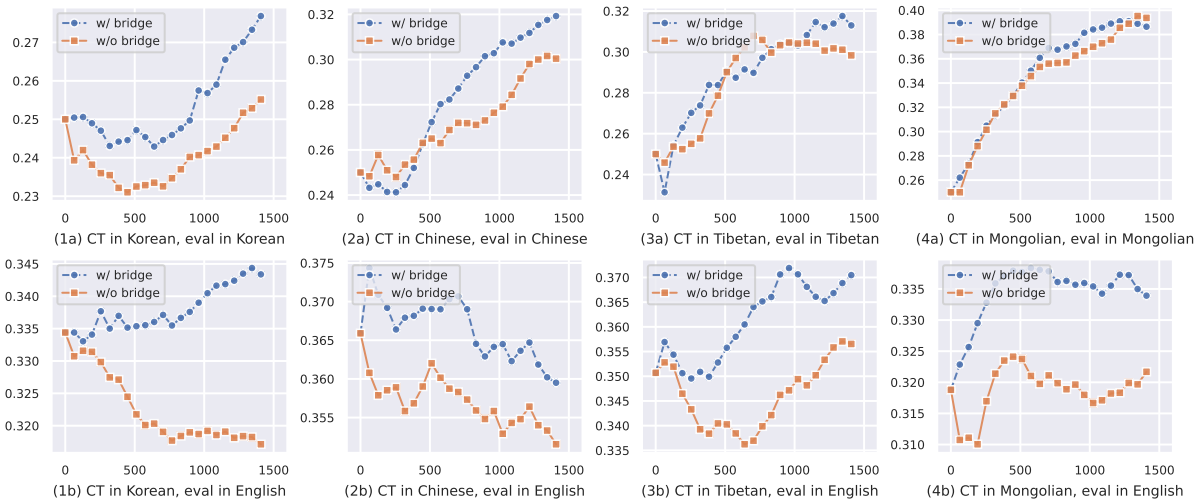


Figure 3: Accuracy on the non-Anglophonic cultural questions with different continual pretraining (CT) steps under different settings. We use EMA smoothing when plotting the figures, with the weight set to 0.8.

tional Ethnic Affairs Commission, 2009), a comprehensive resource on the history and customs of ethnic minorities in China. Specifically, given each paragraph of the book, we ask GPT-4o to generate multi-choice cultural questions, which are then verified and translated by native speakers. In total, we collect between 276 and 598 questions for each culture. See details in Appendix B.

Results and Analyses In Figure 3, we illustrate how the accuracy of probing questions related to non-Anglophonic cultures evolves throughout the continual pretraining process[2]. When the evaluation questions are written in non-English languages (the first row in Figure 3), the gaps between the two settings, one with bridges (blue lines with circular markers) and the other without (orange lines with square markers), can be an estimation of knowledge transferred from English, as opposed to knowledge acquired through continual pretraining. For relatively high-resource languages (Korean and Chinese), we observe notable English-to-non-English transfer (1a and 2a in Figure 3), while for low-resource languages (Tibetan and Mongolian), such transfer is less evident (3a and 4a in Figure 3).

Regarding the transfer from non-English to English (the second row in Figure 3), we observe clear performance gaps between the settings with and without bridges for all four cultures, indicating that the knowledge of non-Anglophonic cultures flows into English. Notably, for most languages (1b, 3b, and 4b in Figure 3), this transfer appears to make up for the forgetting of English abilities due to non-English continual pretraining: the performance on English-version questions in the with-bridge settings continues to improve.

[2]Our main experiments study the transfer of non-Anglophonic cultures. We additionally conduct experiments on the transfer of knowledge exclusively appearing in the Anglophonic culture. See Appendix D for details.

4 Inspection into Training Data

We hypothesize that the cross-lingual transfer of cultural knowledge is largely influenced by its frequency of occurrence in the training corpus, which has been identified as an important factor in monolingual knowledge acquisition (Chang et al., 2024; Li et al., 2025). Specifically, cultural knowledge that appears more frequently in the training corpus of one language is more likely to be transferred to another language. We test this hypothesis by estimating the number of occurrences within our training data for each cultural knowledge piece.

Setup For each cultural probing question, we first retrieve the 50 most relevant documents from the corpus. We then employ Llama-3.1-70B (Dubey et al., 2024) to assess whether each retrieved document entails the cultural knowledge in the question. Given the substantial differences in corpus sizes between English and non-English languages, we introduce the metric of **cultural density**, normalizing the count of knowledge appearance by the total number of documents in the corpus. See Appendix C.3 for implementation details.

Results As shown in Table 1, for cultures associated with low-resource languages (Tibetan and Mongolian), their densities in non-English corpora are one order of magnitude higher than those in

	EN Corp.	Non-EN Corp.
Koreans	2.86e-7	5.21e-7
Han Chinese	2.97e-7	2.84e-7
Tibetans	1.49e-7	9.19e-6
Mongols in China	1.55e-7	3.72e-6

Table 1: The cultural densities of different cultures in the English and non-English corpora.

the English corpus. Thus, the knowledge transfer from non-English to English is more evident than that in the reverse direction. Meanwhile, for the cultures of higher-resource languages (Korean and Chinese), the cultural densities between English and non-English corpora are in the same order of magnitude. Accordingly, the extent of cross-lingual transfer is similar for both transfer directions.

Beyond this culture-level comparison, we look at individual data points, examining the subset of cultural knowledge that is successfully transferred between English and non-English (see Appendix C.4 for selection criteria). We find that these transferred knowledge pieces occur more frequently in the corpus than average. For cultural knowledge transferred from English to non-English, the average occurrence in the English corpus is 9.0, significantly higher than the overall average of 4.2. Similarly, for cultural knowledge transferred from non-English to English, the average occurrence in the non-English corpus is 4.7, compared to the general average of 2.2. These findings further support our frequency-based hypothesis.

5 Conclusion

We investigate cross-lingual cultural knowledge transfer during LLM language adaptation. With an interpretable research framework, we observe bidirectional transfer for high-resource languages but an asymmetric pattern for low-resource languages. We propose a frequency-based hypothesis to explain this phenomenon, with evidence from the analysis of pretraining corpora. We hope that our study could inspire more efforts to uncover the dynamics of knowledge transfer and improve the cultural awareness of LLMs, especially for low-resource languages.

Limitations

Model Scale We only use 0.5B models for experiments because our experiments involve pretraining for 16 different settings, which requires considerable computational resources. We select our model size as a compromise between the informativity of results and computational cost.

Coverage of Cultures To control the factors of cross-lingual transfer in the experiments, we need to use non-Indo-European languages employing non-Latin writing systems, which exhibit greater typological divergence from English and have little lexical overlap with English. This strict requirement significantly narrows the range of suitable cultures for study. Additionally, the collection, verification, and translation of cloze-style cultural questions demand substantial human effort, as existing cultural evaluation datasets are typically monolingual and open-ended, which are not suitable for our study.

Considering these constraints, we select four culturally representative cases, ensuring a balance between well-represented and underrepresented cultures for a more comprehensive analysis. Notably, for the two underrepresented cultural communities, Tibetans and Mongols in China, no existing NLP studies on cultural analysis have considered them. Our evaluation dataset thus provides a valuable resource for investigating these low-resource languages and their cultural representations.

Factors of Cross-Lingual Transfer In our research framework, we try our best to control the factors of cross-lingual transfer for explainable experiments, including language similarities, lexical overlaps, and parallel co-occurrences, which have been identified as primary contributors to cross-lingual transfer (Radford et al., 2019; Blevins and Zettlemoyer, 2022; Philippy et al., 2023). Although several studies find that model architectures (Dufter and Schütze, 2020) and pretraining objectives (Liu et al., 2020) are related to cross-lingual transfer, their effects have been extensively validated at scale. Also, these model-related variables remain constant across different settings in our experiments, minimizing their influence on our findings.

Retrieval Systems Our analysis of the training data involves retrieving documents relevant to the studied knowledge pieces, a common practice in research on knowledge acquisition (Kandpal et al., 2023; Li et al., 2025; Wang et al., 2025). However, retrieval systems are not guaranteed to be perfect. To mitigate potential inaccuracies, we carefully design and optimize retrieval systems for each language, ensuring robust performance and reliable results. See details in Appendix C.3.

Acknowledgements

This work is supported in part by NSFC (62161160339) and Beijing Science and Technology Program (Z231100007423011). We thank Xiao Liu, Mingxu Tao, and the anonymous reviewers for their valuable feedback. We also thank all the annotators who contributed to the dataset construction. For any correspondence, please contact Yansong Feng.

Ethical Considerations

A potential concern when collecting cultural questions from existing sources, such as online corpora or books, is the risk of misrepresenting cultures, particularly for underrepresented cultures using low-resource languages. To address this issue, we ensure that all cultural probing questions are verified by native speakers from the respective communities. This process guarantees that our dataset contains no inaccurate or offensive descriptions of their cultures. We discuss this process in detail in Appendix B.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explains the cross-lingual capabilities of English pretrained models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3563–3574, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. 2024. How do large language models acquire factual knowledge during pretraining? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, et al. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *arXiv preprint arXiv:2412.04261*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Philipp Dufter and Hinrich Schütze. 2020. Identifying elements essential for BERT's multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4423–4437, Online. Association for Computational Linguistics.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2024. Bertaqa: How much do language models know about local culture? *arXiv preprint arXiv:2406.07302*.

Yi Fung, Ruining Zhao, Jae Doo, Chenkai Sun, and Heng Ji. 2024. Massively multi-cultural knowledge acquisition & lm benchmarking. *arXiv preprint arXiv:2402.09369*.

Daniel Hershcovich, Stella Frank, Heather Lent, Miryam de Lhoneux, Mostafa Abdou, Stephanie Brandl, Emanuele Bugliarello, Laura Cabello Piqueras, Ilias Chalkidis, Ruixiang Cui, Constanza Fierro, Katerina Margatina, Phillip Rust, and Anders Søgaard. 2022. Challenges and strategies in cross-cultural NLP. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6997–7013, Dublin, Ireland. Association for Computational Linguistics.

Peng Hu, Sizhe Liu, Changjiang Gao, Xin Huang, Xue Han, Junlan Feng, Chao Deng, and Shujian Huang. 2024. Large language models are cross-lingual knowledge-free reasoners. *arXiv preprint arXiv:2406.16655*.

Kaiyu Huang, Fengran Mo, Hongliang Li, You Li, Yuanchi Zhang, Weijian Yi, Yulong Mao, Jinchen Liu, Yuzhuang Xu, Jinan Xu, et al. 2024. A survey on large language models with multilingualism: Recent advances and new frontiers. *arXiv preprint arXiv:2405.10936*.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR.

Nora Kassner, Philipp Dufter, and Hinrich Schütze. 2021. Multilingual LAMA: Investigating knowledge in multilingual pretrained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3250–3258, Online. Association for Computational Linguistics.

Eunsu Kim, Juyoung Suk, Philhoon Oh, Haneul Yoo, James Thorne, and Alice Oh. 2024. CLIcK: A benchmark dataset of cultural and linguistic intelligence

in Korean. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3335–3346, Torino, Italia. ELRA and ICCL.

Huihan Li, Arnav Goel, Keyu He, and Xiang Ren. 2025. Attributing culture-conditioned generations to pretraining corpora. In *The Thirteenth International Conference on Learning Representations*.

Huihan Li, Liwei Jiang, Nouha Dziri, Xiang Ren, and Yejin Choi. 2024. CULTURE-GEN: Revealing global cultural perception in language models through natural language prompting. In *First Conference on Language Modeling*.

Peiqin Lin, André FT Martins, and Hinrich Schütze. 2024. A recipe of parallel corpora exploitation for multilingual large language models. *arXiv preprint arXiv:2407.00436*.

Chen Cecilia Liu, Iryna Gurevych, and Anna Korhonen. 2024. Culturally aware and adapted nlp: A taxonomy and a survey of the state of the art. *arXiv preprint arXiv:2406.03930*.

Chi-Liang Liu, Tsung-Yuan Hsu, Yung-Sung Chuang, and Hung-Yi Lee. 2020. A study of cross-lingual ability and language-specific information in multilingual bert. *arXiv preprint arXiv:2004.09205*.

and National Ethnic Affairs Commission. 2009. *Ethnic Minorities in China (Revised Edition)*. 5. People's Publishing House.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846.

Siddhesh Pawar, Junyeong Park, Jiho Jin, Arnav Arora, Junho Myung, Srishti Yadav, Faiz Ghifari Haznitrama, Inhwa Song, Alice Oh, and Isabelle Augenstein. 2024. Survey of cultural awareness in language models: Text and beyond. *arXiv preprint arXiv:2411.00860*.

Siddhesh Pawar, Junyeong Park, Jiho Jin, Arnav Arora, Junho Myung, Srishti Yadav, Faiz Ghifari Haznitrama, Inhwa Song, Alice Oh, and Isabelle Augenstein. 2025. Survey of cultural awareness in language models: Text and beyond. *Computational Linguistics*, pages 1–96.

Fred Philippy, Siwen Guo, and Shohreh Haddadan. 2023. Towards a common understanding of contributing factors for cross-lingual transfer in multilingual language models: A review. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5877–5891, Toronto, Canada. Association for Computational Linguistics.

Jirui Qi, Raquel Fernández, and Arianna Bisazza. 2023. Cross-lingual consistency of factual knowledge in multilingual language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10650–10666, Singapore. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Weiyan Shi, Ryan Li, Yutong Zhang, Caleb Ziems, Raya Horesh, Rogério Abreu de Paula, Diyi Yang, et al. 2024. Culturebank: An online community-driven knowledge base towards culturally aware language technologies. *arXiv preprint arXiv:2404.15238*.

Xinyi Wang, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang. 2025. Generalization v.s. memorization: Tracing language models' capabilities back to pretraining data. In *The Thirteenth International Conference on Learning Representations*.

Xiangpeng Wei, Haoran Wei, Huan Lin, Tianhao Li, Pei Zhang, Xingzhang Ren, Mei Li, Yu Wan, Zhiwei Cao, Binbin Xie, et al. 2023. Polylm: An open source polyglot large language model. *arXiv preprint arXiv:2307.06018*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Jiacheng Ye, Xijia Tao, and Lingpeng Kong. 2023. Language versatilists vs. specialists: An empirical revisiting on multilingual transfer ability. *arXiv preprint arXiv:2306.06688*.

Da Yin, Hritik Bansal, Masoud Monajatipoor, Liunian Harold Li, and Kai-Wei Chang. 2022. GeoMLAMA: Geo-diverse commonsense probing on multilingual pre-trained language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2039–2055, Abu

Dhabi, United Arab Emirates. Association for Computational Linguistics.

Fei Yuan, Yinquan Lu, Wenhao Zhu, Lingpeng Kong, Lei Li, Yu Qiao, and Jingjing Xu. 2023. Lego-MT: Learning detachable models for massively multilingual machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11518–11533, Toronto, Canada. Association for Computational Linguistics.

Chen Zhang, Mingxu Tao, Quzhe Huang, Jiuheng Lin, Zhibin Chen, and Yansong Feng. 2024. MC^2: Towards transparent and culturally-aware NLP for minority languages in China. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8832–8850, Bangkok, Thailand. Association for Computational Linguistics.

Xin Zhao, Naoki Yoshinaga, and Daisuke Oba. 2024. Tracing the roots of facts in multilingual language models: Independent, shared, and transferred knowledge. *arXiv preprint arXiv:2403.05189*.

A Information of Studied Cultures

In Table 2, we briefly introduce the four communities studied in this work, including regions, populations, languages, and writing systems.

B Collection of Cultural Questions

Collection of Questions for Koreans We use the data from CLIcK (Kim et al., 2024), a benchmark evaluating LLMs' knowledge of the Korean language and culture. Each data point in it is a multi-choice question.

From the *culture* subset of CLIcK, we select the data points where the question length does not exceed 300 characters and the choice length does not exceed 30 characters. In this way, we filter out the lengthy questions requiring reading comprehension, which are not suitable for the evaluation of small models such as the 0.5B ones in our experiments.

As the questions in CLIcK end with a question mark, we convert them into cloze-style questions with GPT-4o. A cloze-style question ends with a period and has a blank in itself, where the correct answer fits, similar to the format of MLAMA (Kassner et al., 2021). We further filter the questions that do not meet this requirement after conversion. These questions are further translated from Korean to English by GPT-4o. We manually verify the translation results. Our use of CLIcK is consistent with its intended use.

Collection of Questions for Han Chinese, Tibetans, and Mongols in China For the other three cultures in China, we construct probing questions from scratch, as there is no suitable data available.

We excerpt the chapter describing each group from the book 《中国少数民族》 (*Ethnic Minorities in China*; National Ethnic Affairs Commission, 2009), a book written in Chinese describing the history and custom of China's ethnic minorities. We then ask GPT-4o to generate multi-choice cultural questions based on each paragraph of the chapter. To facilitate the generation of distracting answers to the questions, we additionally provide GPT-4o with paragraphs that discuss the same topic but describe other ethical groups.

After generation, we first ask native speakers from each cultural community to check whether the questions faithfully reflect their cultures. For Chinese and Tibetan, all the questions pass the check, while for Mongolian, 2% of the generated questions are marked as misrepresenting their cultures.

The collected questions are translated from Chinese into English by GPT-4o. We manually verify the translation results. For the questions related to Tibetan and Mongolian cultures, we ask native speakers to translate them from Chinese into Tibetan and Mongolian, respectively.

The annotators are recruited from universities, who are native speakers of the low-resource languages and proficient in Chinese. They are informed of how the collected data will be used. They are paid approximately $1 for translating a question, which is fair given the participants' demographic.

Data Statistics In total, we collect 598 questions for Koreans, 328 questions for Han Chinese, 268 questions for Tibetans, and 552 questions for Mongols in China. See other statistics in Table 3.

C Implementation Details

C.1 Continual Pretraining

Data Regarding English pretraining, we use the whole Wikipedia[3] for pretraining, which amounts to approximately 5B tokens. We remove all the non-Latin characters in it, ensuring that during pretraining, the model do not see contents written in the four non-English languages of our study.

For the continual pretraining of four non-English languages, we use both monolingual and parallel corpora. Their ratio is 1:1. Regarding monolingual pretraining corpora, we use Wikipedia for English, Chinese, and Korean. For Mongolian and Tibetan, we use MC^2 (Zhang et al., 2024), as the sizes of Wikipedia for these languages are too small for pretraining. Regarding parallel corpora, we use the data from Lego-MT (Yuan et al., 2023) for Chinese and Korean. For Mongolian and Tibetan, we use the parallel data from the National Basic Science Data Center of China[4]. For the setting with cross-lingual transfer bridges, we use the templates to concatenate each pair of parallel sentences into a document. For example, the template for concatenating English-Chinese parallel sentences is `English: {english_setentece} Chinese: {chinese_sentence}`.

[3] https://huggingface.co/datasets/wikimedia/wikipedia

[4] https://cstr.cn/16666.11.nbsdc.vtfshbjs

	Region	Population	Native Language	ISO 639-1	Writing System
Koreans	South Korea	52M	Korean	ko	Hangul
Han Chinese	China	1.3B	Chinese	zh	Chinese characters
Tibetans	China	7.1M	Tibetan	bo	Tibetan script
Mongols in China	China	6.3M	Mongolian	mn	Traditional Mongolian script

Table 2: Information of the cultural communities and their languages studied in our paper.

	ko	zh	bo	mn
Number of Questions	598	328	268	276
Avg. Len. of EN Questions	16.3	22.8	18.0	19.7
Avg. Len. of EN Options	4.1	3.6	3.0	3.2
Avg. Len. of Non-EN Questions	20.5	17.3	91.3	111.6
Avg. Len. of Non-EN Options	5.1	2.6	14.6	20.5

Table 3: Statistics of the cultural probing questions. The average lengths are measured by the tokens produced by the tokenizer of Qwen-2.5-0.5B.

Hyperparameters We use Deepseed[5] for training. The model is trained for up to 1500 steps using a batch size of 0.5M tokens, a learning rate of 1e-4, and a warmup ratio of 0.01 on eight A100 GPUs. A training step takes approximately 18 seconds.

C.2 Evaluation

The probing questions are in a cloze style, each paired with four candidate answers. We put each answer candidate into the blank in the question and calculate the perplexity of the sentence. The candidate with the lowest perplexity is chosen as the final prediction. We use the evaluation scripts from Qi et al. (2023).

C.3 Knowledge Retrieval

We adopt different methods for the knowledge retrieval of high-pressure and low-resource languages, due to their differences in the availability of mature retrieval techniques.

Retrieval for High-Resource Languages For high-resource languages such as English, Chinese, and Korean, we adopt competitive dense retrieval models that are available for these languages. We use `BAAI/bge-small-en-v1.5`[6] (Xiao et al., 2023) for English, `BAAI/bge-small-zh-v1.5`[7] for Chinese, and `upskyy/bge-m3-korean`[8] (Chen et al., 2024) for Korean. Before retrieval, the documents in the corpora are split into chunks no longer than 5,000 characters. After retrieval, we use Llama-3.1-70B (Dubey et al., 2024) to determine whether a retrieved document can entail each query.

Retrieval for Low-Resource Languages For low-resource languages, i.e. Tibetan and Mongolian, no dense retrieval models are available, and current LLMs perform poorly on the NLI task in these languages. Therefore, we mainly adopt lexical-based retrieval and translate queries and documents into high-resource languages for NLI.

Specifically, we first retrieve the top 50 documents using BM25 (Robertson et al., 2009) method, a language-agnostic retrieval algorithm. Before retrieval, the documents are split into chunks no longer than 5,000 characters. Then, we translate both queries and retrieved documents into Chinese using `NLLB-200-3.3B`[9] (NLLB Team et al., 2024), We then use Llama-3.1-70B (Dubey et al., 2024) to determine the entailment relationship between each query and its 50 retrieved documents.

C.4 Criterion of Successfully Transferred Instances

In our instance-level analysis, we adopt a strict criterion to identify the knowledge pieces successfully transferred across languages.

For a knowledge piece successfully transferred from non-English to English, its corresponding cultural question should (1) be incorrectly answered in the English version before continual pretraining (CT), (2) be correctly answered in the non-English version by the last three CT checkpoints of the no-bridge setting, and (3) be correctly answered in the English version by the last three CT checkpoints of the bridged setting.

For a knowledge piece successfully transferred from English to non-English, its corresponding cultural question should (1) be correctly answered in the English version before CT, (2) be incorrectly

[5] https://github.com/microsoft/DeepSpeed

[6] https://huggingface.co/BAAI/bge-small-en-v1.5

[7] https://huggingface.co/BAAI/bge-small-zh-v1.5

[8] https://huggingface.co/upskyy/bge-m3-korean

[9] The model is further finetuned with Chinese-Tibetan and Chinese-Mongolian parallel data to enhance its translation capabilities.

answered in the non-English version by the last three CT checkpoints of the no-bridge setting, and (3) be correctly answered in the non-English version consistently by the last three CT checkpoints of the bridged setting.

D Transfer of Knowledge from Anglophonic Cultures

Data Collection It is untrivial to construct cultural questions for *English culture* as English is the native language of many countries or communities, whose cultures may differ greatly. As a workaround, we study the information of celebrities from the core Anglosphere (Australia, Canada, New Zealand, the United Kingdom, and the United States). This kind of information is more likely to appear in the English corpus than in non-English corpora.

Specifically, we sample 1,000 persons from Wikidata who meet the following requirements: (1) their nationality is in the core Anglosphere, (2) they have only English Wikipedia pages and do not have pages in the four languages of our study (Chinese, Korean, Mongolian, and Tibetan). We then obtain their attributes, including nationality, date of birth, place of birth, occupation, and educational background. We further ensure that their names and attributes only appear in the English corpus, and not in the corpus of other languages.

These person-attribute pairs are then converted to multi-choice questions using templates in different languages, such as `{person} was born in {birthplace}` and `{person}`出生在`{birthplace}`. We directly use the person names and their attributes in their original English forms, as translating them requires substantial human labor. In total, we collect 3,073 such questions.

Experiments and Results We investigate whether Anglophonic cultural knowledge acquired during English pretraining can be transferred to other languages after continual pretraining. Following the experimental setup outlined in Section 2, we compare two settings: one with cross-lingual transfer bridges and one without. After 1,500 steps of continual pretraining, we evaluate the models using non-English versions of Anglophonic cultural questions.

Our results show that the setting with transfer bridges achieves, on average, 20% higher accuracy compared to the setting without bridges, indicating that Anglophonic cultural knowledge can be transferred to non-English languages. This finding is also in line with our frequency-based hypothesis in Section 4: Anglophonic cultural knowledge appears more frequently in English pretraining data than in non-English data, making it more easily transferable to other languages.

Leveraging Human Production-Interpretation Asymmetries to Test LLM Cognitive Plausibility

Suet-Ying Lam[1*], **Qingcheng Zeng**[2*], **Jingyi Wu**[2], **Rob Voigt**[2]

[1]UMass Amherst, [2]Northwestern University

Abstract

Whether large language models (LLMs) process language similarly to humans has been the subject of much theoretical and practical debate. We examine this question through the lens of the production-interpretation distinction found in human sentence processing and evaluate the extent to which instruction-tuned LLMs replicate this distinction. Using an empirically documented asymmetry between pronoun production and interpretation in humans for implicit causality verbs as a testbed, we find that some LLMs do quantitatively and qualitatively reflect human-like asymmetries between production and interpretation. We demonstrate that whether this behavior holds depends upon both model size-with larger models more likely to reflect human-like patterns and the choice of meta-linguistic prompts used to elicit the behavior. Our codes and results are available here.

1 Introduction and Background

The extent to which large language models (LLMs) are "cognitively plausible," that is, replicate human-like behaviors in language processing, has been the subject of ongoing debate (Dentella et al., 2023; Hu et al., 2024; Futrell and Mahowald, 2025; Kuribayashi et al., 2025). Existing research on the linguistic capabilities of LLMs has predominantly focused on their performance in language interpretation, e.g., pragmatic understanding (Hu et al., 2023), sentence acceptability judgment (Warstadt et al., 2020), garden-path effect (Futrell et al., 2019), reference resolution (Lam et al., 2023). In this study, we examine a previously unexplored dimension of cognitive plausibility: whether LLMs reflect human-like distinctions between **production** and **interpretation** in language processing.

Production and interpretation were traditionally treated as two independent processes in human language: for instance, in neurolinguistics the "classic" Lichtheim–Broca–Wernicke model assumes distinct anatomical pathways associated with production and interpretation (see Ben Shalom and Poeppel 2008). While this extreme dichotomy has been rejected recently (see Pickering and Garrod 2013), humans do exhibit different underlying biases in language processing between production and interpretation even in very similar tasks. Whether such distinctions carry over into LLMs is of particular interest when we consider that the fundamental unit of LLM computation is $P(token|context)$, which is inherently ambiguous between production and interpretation and is practically applied towards both types of tasks.

The present study probes into this question using reference processing as a test case. Consider the following examples:

(1) A production task: next-mentioned bias
  a. John infuriated Bill. ... [IC1]
  b. John praised Bill. ... [IC2]

(2) A interpretation task: ambiguous pronoun resolution
  a. John infuriated Bill. He ... [IC1]
  b. John praised Bill. He ... [IC2]

When asked to continue the story in (1), speakers usually describe events that happened to one of the two mentioned characters. This is a production task, and psycholinguistic research has investigated how the preceding context affects the **next-mention bias** of the character, i.e., how likely a character will be referred to in the continued story $P(referent|context)$. In (1-a), 'John' has a higher next-mention bias than 'Bill', because he is the implicit cause of the event. Verbs like 'infuriate' are therefore called the subject-biased im-

*Both authors contributed equally. Correspondence to qcz@u.northwestern.edu

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 158–171
July 27 - August 1, 2025

plicit causality (IC1) verbs. In contrast, 'Bill' has a higher next-mention bias than 'John' in (1-b), because 'Bill' implies an implicit cause for 'John' to praise him. Verbs like 'praise' are therefore called object-biased implicit causality (IC2) verbs (e.g., Stevenson et al. 1994).

A similar implicit causality bias can also be found in (2). Although participants are asked to perform the same sentence continuation task as in (1), note that an additional interpretation element is involved here: since an additional pronoun is provided after the context sentence, participants must first resolve the ambiguous pronoun "he" $P(referent|pronoun)$ before providing a reasonable continuation.

Studies have shown that human participants were more likely to resolve the ambiguous pronoun to the subject 'John' than the object 'Bill' with an IC1 verb. That is, 'John' has a higher interpretation bias than 'Bill' under this condition. This interpretation bias flips when the verb is an IC2 verb: 'Bill' were more likely to be the antecedent of the ambiguous pronoun 'he' than 'John' in (2-b) (e.g., Crinean and Garnham 2006).

Given the additional interpretation element, henceforth we refer to the sentence continuation task with a given pronoun such as (2) as an **interpretation task** for measuring pronoun interpretation bias; and the one without a given pronoun such as (1) as the **production task** for measuring next-mentioned bias, despite the fact that the sentence continuation task is in principle a production task.

Crucially, psycholinguistic research has revealed an asymmetry between these two biases. In the interpretative case of pronoun interpretation bias, humans are robustly more likely to show a preference for the subject than in the production case of next-mention bias cross-linguistically (English: Rohde and Kehler 2014; Mandarin: Zhan et al. 2020; Lam and Hwang 2024; German: Patterson et al. 2022; Catalan: Mayol 2018). That is, participants were more likely to resolve an ambiguous pronoun towards the subject than choose the subject as the next referent, despite the same context.

For humans, this extra subject bias in interpretation comes from the bias of using pronouns for subject antecedent, i.e., $P(he|subject)$. That is, when they see an ambiguous pronoun, they do not only consider the next-mention bias of which antecedent is more likely to be mentioned, but also why a pronoun is used. It is unknown whether and how LLMs can handle this difference, as they do not generate such a conditional probability based on the choice of next referent instead of the context.

We therefore probe this dimension of LLM cognitive plausibility using this task, asking (1) whether the IC verb-type effect is reflected by LLMs in both production and interpretation; and (2) whether a human-like asymmetry between the two biases exists. This is our first set of questions: do LLMs show human-like interpretation and production biases, and if so under what conditions? Do human-like effects scale with parameter count?

Evaluating LLM in metalinguistic prompts Hu and Levy (2023) demonstrated that direct probability-based measures in general outperformed meta-linguistic prompting in assessing plausibility and syntactic processing tasks. However, not all language processing tasks can be effectively quantified using probability-based measures, and for some tasks meta-linguistic prompts are the only possible method to measure processing. This is exactly our case: in the ambiguous pronoun resolution task, the bias towards the subject 'John' or the object 'Bill' is represented by the same term, i.e., $P(he|context)$. Metalinguistic prompting is thus necessary to elicit meaningful results.

One might expect LLM performance to vary within different metalinguistic prompts, but it is unclear which type of metalinguistic prompts would perform better. Although Hu and Levy (2023) demonstrated that metalinguistic prompts that are more similar to the direct probability baseline perform better, we would like to explore whether probability measurements obtained via metalinguistic prompting also provide greater reliability. This constitutes the second aim of this paper: across different metalinguistic prompting strategies, which elicit more human-like language processing behaviors?

Our findings are summarized below: (1) in the most cases, LLMs cannot capture the difference between the production and the interpretation task in reference processing; if any, the asymmetry is limited to certain meta-linguistic prompts and rarely reach the magnitude found in human participants; and (2) the choice of meta-linguistic prompts matters in evaluating LLMs: most LLMs perform in a more human-like manner with one specific prompt that is unrelated to probability measure, namely the Yes/No prompt (details explained in Section 2.2).

2 Methodology

2.1 Stimuli

We constructed stimuli in the frame of "[Character A] IC-verb [Character B]" without pronouns for the next-mention production task (as in (1)), and with a pronoun with an ambiguous referent for the pronoun resolution task (as in (2)). We selected 137 IC1 verbs and 134 IC2 verbs from the original study that found the production-interpretation asymmetry (Rohde and Kehler, 2014), and the English IC verb corpus (Ferstl et al., 2011). For each verb, we heuristically created two items by assigning a pair of male names and a pair of female names randomly selected from 13 unambiguously male and 13 unambiguously female names from Rohde (2008). The congruence of the gender of the characters ensures the ambiguity of the pronoun. This results in 541 items in each task.

2.2 Models and Metalinguistic Prompting

Our experiments evaluate four representative LLMs from open-source LLMs ranging from 8B to 70B and one proprietary LLM: `LLaMA3.1Instruct-8B` (Grattafiori et al., 2024), `QWen2.5Instruct-32B` (Qwen et al., 2025), `LLaMA3.3Instruct-70B` (Grattafiori et al., 2024), and `GPT-4o` (OpenAI et al., 2024). We focus on instruction-tuned model [1] as they allow the effective use of metalinguistic prompting, and varying parameter counts also allow us to assess the impact of model scaling on human-like language processing behavior. Specifically, we constantly use greedy decoding in our generation. We employ four metalinguistic prompt strategies to assess LLM behavior:

(i) Binary choice prompting: The model is prompted to select between subject and object.

(ii) Continuation prompting: The model is instructed to extend the sentence by continuing with either the subject or the object.

(iii) Yes/No prompting: The model is asked whether the following sentence (or the existing pronoun) will begin with the *subject*, requiring a binary response.

(iv) Yes/No probability prompting: Similar to (iii), but instead of a categorical response, we extract the probability assigned to the *Yes* token as a quantitative measure.

Three authors of the paper manually verified all model outputs to confirm subject/object choice and exclude ambiguous, nonsensical, plural responses, and responses with repeated pronoun in the interpretation task. Table 2 reported the distribution of the excluded responses over the 1082 responses of each model in the two tasks.

Table 1: The distribution of excluded responses in continuation prompting

	Ambiguous	Non-sensical	Plural	Repeated pronoun
LLaMA-8B	54	21	17	0
LLaMA-70B	35	9	0	0
Qwen	17	6	0	0
GPT-4o	22	1	8	5

The specific prompts used in our experiments are provided in Appendix A, and the details of the exclusion criteria in Appendix B.1.

2.3 Evaluation

We evaluated model behavior from two perspectives. We first consider whether the IC verb type effect and the production-interpretation asymmetry exists in LLMs, which we both visualize to observe broad trends and verify with statistical tests like those run in human experiments. We then consider the magnitude of the effect found in LLMs, as psycholinguistic research has found that language models often fail to replicate the magnitude of effects found in human participants even when the directionality is similar. Below we only report effects that are verified by statistical evidence, the details of which be found in Appendix B.

3 Results

Result #1: Implicit causality biases are sometimes replicated by LLMs, depending on prompts and models. Figure 1 presents the task performance across different metalinguistic prompts for each model. The implicit causality (IC) verb effect — where subjects are chosen more frequently after IC1 verbs than IC2 verbs — was observed in at least one metalinguistic prompt for all models, though the strength and consistency of this effect varied by both model and prompt type.

For LLaMA models, the IC verb effect was present in both production and interpretation when using Yes/No and Yes/No probability prompts, indicating a broader sensitivity to IC biases across tasks. In contrast, the IC verb effect was limited

[1] We also experiment with base LLMs. However, even for continuation prompting, base LLMs could not follow instructions reliably.

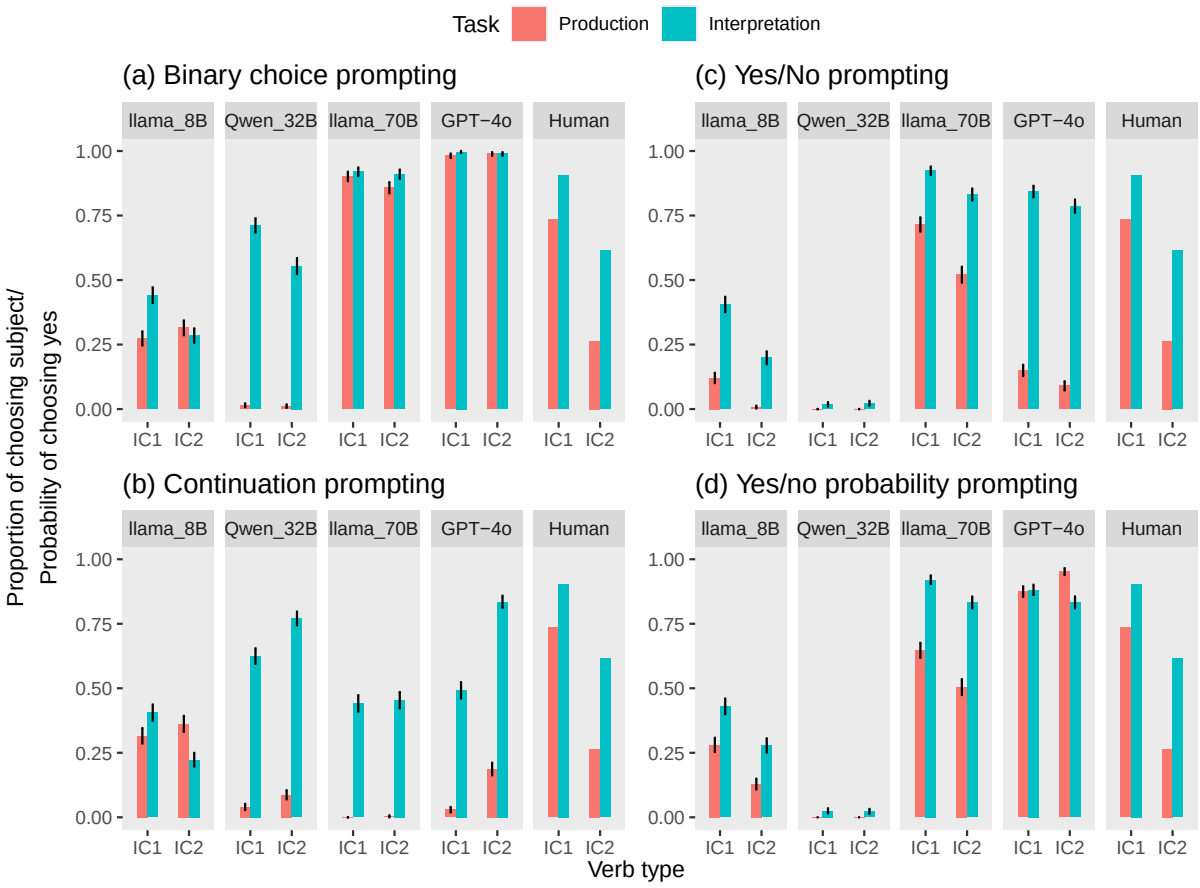


Figure 1: Model behavior as proportion of subject/yes choice as the antecedent by prompting strategy. Human behavior – rightmost facet in each subfigure for reference, from Rohde and Kehler (2014) – tends to reflect higher subject bias for IC1 over IC2 verbs, but with asymmetry between production and interpretation tasks.

to Yes/No prompting for GPT-4o and even more restricted for `Qwen`, in which the effect was only observed in interpretation and limited to binary choice prompt. This suggests that the model's sensitivity to IC verbs may depend on task framing.

Result #2: The production-interpretation asymmetry is limitedly captured. Recall that human participants are more subject-biased in interpretation than in production. This difference is only elicited in LLaMA models using Yes/No and Yes/No probability prompts, and GPT-4o using Yes/No prompt and continaution prompt. `Qwen` was entirely unable to capture this asymmetry because it failed to predict the IC verb effect in production with all four metalinguistic prompts.

Another unexpected production-interpretation asymmetry is that LLMs generally are less likely to capture the IC verb bias in production than in interpretation. For instance, while the `LLaMA-8B` model was able to predict an IC verb effect in interpretation when using binary choice and continuation prompts, it predicted a reverse verb type effect in production using these prompts. The same pattern can also be seen in GPT-4o using Yes/No probability prompts. This unexpected asymmetry is even clearer in the performance of the `Qwen` model, in which the IC verb effect is only found in interpretation. This suggests that LLMs are generally unable to recognize the difference between production and interpretation like humans.

Result #3: LLMs do not align with human behavior in effect magnitude. Although there was a verb type effect and a production/interpretation task effect on LLMs' responses, the magnitude of these effects is different from humans even in the most similar case: with Yes/No prompting, `LLaMA-70B` predicted an IC verb effect difference of 19.4% in the production task and 9.2% in the interpretation task. Yet, these two differences were 47.2% and 28.8% respectively in human participants (Rohde and Kehler, 2014). In other words, even in the best scenario, LLMs underestimate the magnitude of the IC verb effect.

Surprisingly, the magnitude of the production-

interpretation asymmetry predicted by LLMs is well-aligned with humans in one case: in human participants, this asymmetry was 16.8% and 35.2% respectively for IC1 and IC2 verbs. With Yes/No prompting, LLaMA-70B predicted a 20.8% and 31% difference respectively for the two verb types. Yet, this seems to be an isolated case. Among the six cases (two LLaMA models in Yes/No and Yes/No probability promptings; GPT-4o in Yes/No and continaution promptings) that exhibit a trend of the production-interpretation asymmetry, this is the only scenario that the magnitude of the asymmetry is similar between LLM and humans. We therefore do not tend to generalize that LLM aligns with human behavior in terms of effect magnitude.

Result #4: Scaling matters. Overall, LLaMA-70B shows better performance than LLaMA-8B across all four prompts. With Binary choice prompts, LLaMA-8B predicted a reverse IC verb effect in the production task and a reverse production-interpretation asymmetry in IC2 verbs, while both effects were captured by LLaMA-70B. With Yes/No and Yes/No probability prompts, although both LLaMA-8B and LLaMA-70B reflected an IC verb effect and production-interpretation asymmetry, the magnitude in human participants was better approximated by LLaMA-70B.

Result #5: Overall, Yes/No prompts align the best with human performance. Among the four promptings used in our study, most models best align with human performance with the Yes/No prompting. This is the only prompting that LLaMA models and GPT-4o capture the IC verb effect and the production-interpretation asymmetry. However, for QWen, the best performing prompt is the binary choice prompting. This shows that models still differ in their sensitivity to metalinguistic prompts.

4 Discussion

Our study examines the asymmetry between interpretation and interpretation in humans within the context of LLMs, showing that, under specific prompting strategies, certain LLMs can approximate human-like asymmetry. Among the four models we tested, while LLaMA models and GPT-4o performed the best with the Yes/No prompting, Qwen performed the best with the binary choice prompting.

Our findings partially align with Hu and Levy (2023), which suggests that the format of metalinguistic prompts influences the behavior of the model. In our study, the four distinct prompting strategies did lead to considerable variation in LLM performance, highlighting the sensitivity of model responses to prompt formulation. The different preferences for prompting strategies between model families also reinforce the need for extensive experimentation across multiple prompting approaches before evaluating a model using metalinguistic prompts.

Yet, different from their findings, we did not see prompting strategies that are more similar to direct probability measures better capture human preferences. Based on Hu and Levy (2023)'s observations, one might expect that metalinguistic prompts that incorporate explicit probability measures would be more reliably approximate human biases. However, our results show otherwise: although models differ in their best-performing prompts, the Yes/No prompting without probability performed best in three out of the four models in our setting. The probability-based format has never yielded superior alignment with human responses. This indicates that probability cues are not universally beneficial in prompting and that their effectiveness may depend on the specific task or model.

Another worth mentioning pattern in our results is that continuation prompting keeps performing the worst in capturing human-like performance, despite being the task originally performed by human participants. One possible explanation is that instruction-tuned LLMs, having been fine-tuned with instruction data or preference optimization objectives, may develop constrained response patterns (Lin et al., 2023), limiting their flexibility in generating diverse continuations. Additionally, such fine-tuning can reduce conceptual diversity (Murthy et al., 2024), making LLMs less sensitive to implicit biases in language processing. This suggests that continuation prompting may not be well-suited for probing human-like asymmetries in interpretation and production.

In conclusion, our study reveals that while some instruction-tuned LLMs can approximate human-like asymmetries between production and interpretation under specific conditions, their sensitivity and alignment to human behavior are highly dependent on model scale and the choice of metalinguistic prompting strategy. These findings highlight the nuanced and prompt-contingent nature of evaluating cognitive plausibility in LLMs.

Limitations

Our work has two primary limitations. First, our experiments are conducted on three English-centric LLMs (`LLaMA` and `GPT-4o`) and one multilingual-oriented LLM (`Qwen`). This selection may introduce biases into the models' performance, potentially limiting the generalizability of our findings across other LLMs. Second, our study focuses solely on the asymmetry between ambiguous pronoun resolution and production in English, without exploring cross-linguistic variations. Future research could address these limitations by incorporating a more diverse set of LLMs and broadening the scope of languages analyzed.

References

Dale J Barr, Roger Levy, Christoph Scheepers, and Harry J Tily. 2013. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language*, 68(3):255–278.

Dorit Ben Shalom and David Poeppel. 2008. Functional anatomic models of language: assembling the pieces. *The Neuroscientist*, 14(1):119–127.

Marcelle Crinean and Alan Garnham. 2006. Implicit causality, implicit consequentiality and semantic roles. *Language and Cognitive Processes*, 21(5):636–648.

Vittoria Dentella, Fritz Günther, and Evelina Leivada. 2023. Systematic testing of three language models reveals low language accuracy, absence of response stability, and a yes-response bias. *Proceedings of the National Academy of Sciences*, 120(51):e2309583120.

Evelyn C Ferstl, Alan Garnham, and Christina Manouilidou. 2011. Implicit causality bias in english: A corpus of 300 verbs. *Behavior Research Methods*, 43:124–135.

Richard Futrell and Kyle Mahowald. 2025. How linguistics learned to stop worrying and love the language models. *Preprint*, arXiv:2501.17047.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Andrew Gelman, Aleks Jakulin, Maria Grazia Pittau, and Yu-Sung Su. 2008. A weakly informative default prior distribution for logistic and other regression models.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal

Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. 2023. A fine-grained comparison of pragmatic language understanding in humans and language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4194–4213, Toronto, Canada. Association for Computational Linguistics.

Jennifer Hu and Roger Levy. 2023. Prompting is not a substitute for probability measurements in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5040–5060, Singapore. Association for Computational Linguistics.

Jennifer Hu, Kyle Mahowald, Gary Lupyan, Anna Ivanova, and Roger Levy. 2024. Language mod-

els align with human judgments on key grammatical constructions. *Proceedings of the National Academy of Sciences*, 121(36):e2400917121.

Tatsuki Kuribayashi, Yohei Oseki, Souhaib Ben Taieb, Kentaro Inui, and Timothy Baldwin. 2025. Large language models are human-like internally. *Preprint*, arXiv:2502.01615.

Suet-Ying Lam and Heeju Hwang. 2024. Pronoun interpretation is more subject-biased than expected by the bayesian model. *Language, Cognition and Neuroscience*, pages 1–20.

Suet-Ying Lam, Qingcheng Zeng, Kexun Zhang, Chenyu You, and Rob Voigt. 2023. Large language models are partially primed in pronoun interpretation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9493–9506, Toronto, Canada. Association for Computational Linguistics.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2023. The unlocking spell on base llms: Rethinking alignment via in-context learning. *Preprint*, arXiv:2312.01552.

Laia Mayol. 2018. Asymmetries between interpretation and production in catalan pronouns. *Dialogue Discourse. 2018 Dec; 9 (2): 1-34*.

Sonia K. Murthy, Tomer Ullman, and Jennifer Hu. 2024. One fish, two fish, but not the whole sea: Alignment reduces language models' conceptual diversity. *Preprint*, arXiv:2411.04427.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick

Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Clare Patterson, Petra B Schumacher, Bruno Nicenboim, Johannes Hagen, and Andrew Kehler. 2022. A bayesian approach to german personal and demonstrative pronouns. *Frontiers in psychology*, 12:672927.

Martin J Pickering and Simon Garrod. 2013. An integrated theory of language production and comprehension. *Behavioral and brain sciences*, 36(4):329–347.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Hannah Rohde. 2008. *Coherence-driven effects in sentence and discourse processing*. University of California, San Diego.

Hannah Rohde and Andrew Kehler. 2014. Grammatical and information-structural influences on pronoun production. *Language, Cognition and Neuroscience*, 29(8):912–927.

Rosemary J Stevenson, Rosalind A Crawley, and David Kleinman. 1994. Thematic roles, focus and the representation of events. *Language and cognitive processes*, 9(4):519–548.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Meilin Zhan, Roger Levy, and Andrew Kehler. 2020. Pronoun interpretation in mandarin chinese follows principles of bayesian inference. *PloS one*, 15(8):e0237012.

A Prompt Template

The following shows the metalinguistic prompt design for LLMs.

System Prompt: You are a helpful assistant.

production-binary-choice-template:
In the following sentence, who is more likely to be the subject of the next sentence? {} or {}? Please ONLY return the name without any explanation or extra words.
Sentence: {}
Answer:

production-yes-no-template:
In the following sentence, judge whether the pronoun of the next sentence will refer to {}. Please ONLY answer with 'Yes' or 'No'.
Sentence: {}

production-continuation-template:
Please reasonably continue the sentence with one of the mentioned characters. You should start a new sentence rather than a clause. Please ONLY return the continuation.
Sentence: {}

interpretation-binary-choice-template:
In the following sentence, who is more likely to be the referent of the pronoun? {} or {}? Please ONLY return the name without any explanation or extra words.
Sentence: {}
Answer:

interpretation-yes-no-template:
In the following sentence, judge whether the pronoun refers to {}. Please ONLY answer with 'Yes' or 'No'.
Sentence: {}

interpretation-continuation-template:
Please reasonably continue the sentence following the pronoun. Please ONLY return the continuation.
Sentence: {}

B Statistical analyses and results

We focus on how the bias of IC verbs (IC1 vs. IC2) and the task (production vs. interpretation) affect the outcomes of LLMs in each type of metalinguistic prompts.

B.1 Data annotation

For binary prompts, we annotated based on the name answered by the LLM. For yes-no prompts, we annotated "yes" as “Subject" .

For continuation prompts, we manually annotated the choice of referent (Subject vs. Object) in the production task and the choice of antecedent (Subject vs. Object) in the ambiguous pronoun resolution task based on the meaning of the generated continuation. For instance, for a sentence like “Nick offended Steve. He decided to apologize and clear the air before things escalated further", we annotated the outcome as “Subject". Ambiguous references (e.g., “Zack divorced Paul. He later moved to a new city to start his life over."), nonsensical continuations (e.g., “Janet wanted Kate. She to join her at the party that night, but Kate had already made other plans."), continuations with a plural antecedent (e.g., “Claire played with Jane. They were building a sandcastle on the beach"), and continuations starting with a repeated pronoun in an interpretation task (e.g., “Claire played with Jane. She... she") were excluded. The distribution of the excluded item has been provided in Table 2 in the main text, repeated below:

Table 2: The distribution of excluded responses in continuation prompting

	Ambiguous	Non-sensical	Plural	Repeated pronoun
LLaMA-8B	54	21	17	0
LLaMA-70B	35	9	0	0
Qwen	17	6	0	0
GPT-4o	22	1	8	5

As can be seen, `LLaMA-8B` made more ambiguous and non-sensical continuations than all other models. Besides, it also provided continuations with “they", which violates the requirement posed in the prompts. This might suggest that scaling affects LLMs' ability to follow the instructions. More models should be evaluated to test this hypothesis.

B.2 Analyses

LLaMA models For the results of LLaMA models, we ran a mixed-effects Bayesian bernoulli regression model using the R package `brms` for the binary outcome resulted from continuation, binary, and yes-no prompts (Subject = 1; Object = 0); and a mixed-effects Bayesian linear regression model for the continuous probability outcome from yes-no probability prompt. Note that the dataset used in statistical model for continuation outcome is slightly different from that used in other models, as some of the responses are excluded, as introduced in Appendix B.1.

Each model was fitted using 4 chains, each with 5000 iterations. The first 1000 were warm-up to calibrate the sampler. This results in 12000 posterior samples. They were all built with fixed predictors of IC verbs (sum-coded: IC1 = 0.5; IC2 = -0.5), task type (sum-coded: interpretation = 0.5; production = -0.5), and their interaction. A maximal random structure justified by design is implemented (Barr et al., 2013). For logistic regression models, we used weakly informative priors, i.e., a Cauchy distribution with a center of 0 and a scale of 2.5 for fixed effects following Gelman et al. (2008), and the default setting of the package for the other parameters. For linear regression models, we used a gaussian distribution with a mean of 0 and a standard deviation of 1 as the weakly informative prior for fixed predictors. When there is an interaction effect, we further ran nested models for pairwise comparison.

The Bayesian statistics framework does not use the p-value. We consider the 95% credible interval (Crl) as the evidence for an effect: if the 95% Crl does not include a zero, i.e., it is all positive or negative, we consider there is evidence for an effect. Below we report the estimate and the 95% Crl for each effect.

QWen For `Qwen`, we only ran analysis for continuation prompts, and limited to the effect of the IC verb only. This is because responses of `Qwen` in the production task is so extreme that no statistical model can be successfully fitted. The settings of the mixed-effects Bayesian bernoulli regression model are the same as those used for LLaMA models.

GPT-4o Like LLaMA models, we ran mixed-effects Bayesian bernoulli regression model for binary outcomes from continuation and yes-no prompts. We did not run analysis for binary prompts because responses of `GPT-4o` were extremely subject-biased in all conditions. As a result, no statistical model can be fitted. We also ran linear regression model for probability outcomes from yes-no probability prompts. The settings of the statistical models are the same as in analyses for LLaMA models.

B.3 Results

We bold the predictor in which the effect is supported the statistical evidence, i.e., the 95% Crl does not contain a zero.

B.3.1 LLaMA-3.1-8B Model

Binary prompting Table 3 shows that there was an interaction effect between the IC verb type and the task type.

Formula: binary $\sim$ verb $*$ task $+ (1 +$ task|itemID$)$			
	Estimate	Est. Error	95% CrI
Intercept	-1.80	0.74	**[-3.78, -0.99]**
verb	0.64	0.48	[-0.12, 1.80]
task	0.67	0.75	[-0.65, 2.47]
verb:task	2.06	0.99	**[0.78, 4.61]**

Table 3: Summary of the Bayesian logistic regression model of LLaMA-3.1-8B model, binary choice prompting.

Nested analyses further reveal that the production-interpretation asymmetry is only found in IC1 verbs (Table 4), and the verb type effect is only found in interpretation (Table 5).

Formula: binary $\sim$ verb/task $+ (1 +$ task|itemID$)$			
	Estimate	Est. Error	95% CrI
Intercept	-1.84	0.76	**[-3.81, -0.99]**
verb	0.63	0.47	[-0.14, 1.76]
verbIC1:task	1.76	0.96	**[0.55, 4.24]**
verbIC2:task	-0.40	0.82	[-2.30, 1.14]

Table 4: Pairwise comparison of the task type effect within IC1 and within IC2 conditions using binary choice prompting in LLaMA-3.1-8B model.

Formula: binary $\sim$ task/verb $+ (1 +$ prompt|itemID$)$			
	Estimate	Est. Error	95% CrI
Intercept	-1.80	0.79	**[-3.85, -0.99]**
task	0.65	0.73	[-0.68, 2.37]
taskProduction:verb	-0.43	0.50	[-1.57, 0.42]
taskInterpretation:verb	1.69	0.89	**[0.61, 4.01]**

Table 5: Pairwise comparison of the IC verb type effect within the production and within the interpretation task using binary choice prompting in LLaMA-3.1-8B model.

Yes-no prompting As shown in Table 6, there was an interaction effect between the verb type and task type for the responses of the model.

Formula: yes_no $\sim$ verb $*$ task $+ (1|$itemID$)$			
	Estimate	Est. Error	95% CrI
Intercept	-4.06	0.61	**[-5.44, -3.04]**
verb	3.12	0.67	**[1.97, 4.61]**
task	4.31	0.70	**[3.12, 5.89]**
verb:task	-1.99	0.93	**[-4.01, -0.39]**

Table 6: Summary of the Bayesian logistic regression model of LLaMA-3.1-8B model, yes-no prompting.

Nested analysis in Table 7 further reveals that the model did give more "yes" (or referring to subject) with IC1 verbs than IC2 verbs (as indicated by the positive intercept of the verb predictor). Also, the production-interpretation asymmetry is found within both IC1 and IC2 verbs, such that the model chose more subjects in interpretation than production.

Formula: yes_no $\sim$ verb/task $+ (1|$itemID$)$			
	Estimate	Est. Error	95% CrI
Intercept	-4.06	0.62	**[-5.50, -3.04]**
verb	3.18	0.70	**[2.00, 4.76]**
verbIC1:task	3.22	0.55	**[2.28, 4.43]**
verbIC2:task	5.45	1.14	**[3.58, 8.04]**

Table 7: Pairwise comparison of the task effect within the IC1 and within the IC2 verbs using yes-no choice prompting in LLaMA-3.1-8B model.

Continuation prompting As shown in Table 8, there was an interaction effect between the verb type and the task type for the responses of the model.

Formula: cont $\sim$ verb $*$ task $+ (1|$itemID$)$			
	Estimate	Est. Error	95% CrI
Intercept	-0.81	0.09	**[-0.99, -0.65]**
verb	0.36	0.16	**[0.05, 0.67]**
task	-0.15	0.14	[-0.44, 0.13]
verb:task	1.14	0.30	**[0.57, 1.73]**

Table 8: Summary of the Bayesian logistic regression model of LLaMA-3.1-8B model, continuation prompting.

Nested analyses in Table 9 show that the verb type effect is only found in interpretation.

Formula: cont $\sim$ task/verb + (1|itemID)			
	Estimate	Est. Error	95% CrI
Intercept	-0.81	0.09	**[-0.99, -0.64]**
task	-0.15	0.14	[-0.44, 0.13]
taskProduction:verb	-0.22	0.21	[-0.64, 0.18]
taskInterpretation:verb	0.93	0.22	**[0.51, 1.37]**

Table 9: Pairwise comparison of the IC verb type effect within the production and within the interpretation task using continuation prompting in LLaMA-3.1-8B model.

The pairwise comparison in Table 10 shows that the production-interpretation asymmetry differs in direction between the two verb types: while interpretation is more subject-biased than production in IC1 verbs, production is more subject-biased than interpretation in IC2 verbs.

Formula: cont $\sim$ verb/task + (1|itemID)			
	Estimate	Est. Error	95% CrI
Intercept	-0.81	0.09	**[-0.99, -0.65]**
verb	0.36	0.15	**[0.06, 0.66]**
verbIC1:task	0.42	0.20	**[0.05, 0.81]**
verbIC2:task	-0.73	0.21	**[-1.14, -0.32]**

Table 10: Pairwise comparison of the task type effect within IC1 and within IC2 verbs using continuation prompting in LLaMA-3.1-8B model.

Yes/No probability prompting As shown in Tabel 11, the model did generate a higher probability of 'Yes' (= subject) following IC1 verbs than IC2 verbs, and in interpretation task than in production task.

Formula: subject_yes_probability $\sim$ verb $*$ task + (1|itemID)			
	Estimate	Est. Error	95% CrI
Intercept	0.28	0.01	**[0.27, 0.29]**
verb	0.15	0.01	**[0.12, 0.18]**
task	0.15	0.01	**[0.13, 0.17]**
verb: task	-0.00	0.02	[-0.03, 0.03]

Table 11: Summary of the Bayesian linear regression model of LLaMA-3.1-8B model, Yes/No probability prompting.

B.3.2 LLaMA-3.3-70B

Binary prompting Table 12 clearly shows that the model only reveals the production-interpretation asymmetry, such that it chose more subject in interpretation than in production. There was no clear evidence for the IC verb type effect.

Formula: binary $\sim$ verb $*$ task + (1|itemID)			
	Estimate	Est. Error	95% CrI
Intercept	14.61	3.61	**[9.24, 23.34]**
verb	1.36	1.42	[-1.19, 4.45]
task	2.21	0.73	**[0.98, 3.84]**
verb:task	-1.24	1.07	[-3.44, 0.77]

Table 12: Summary of the Bayesian logistic regression model of LLaMA-3.3-70B model, binary prompting.

Yes-no prompting Table 13 shows that the model is able to capture the IC verb type effect and the production-interpretation asymmetry, such that it responded 'Yes' (=subject) more for IC1 verbs than IC2 verbs and the interpretation task than the production task.

Formula: yes_no $\sim$ verb $*$ task + (1|itemID)			
	Estimate	Est. Error	95% CrI
Intercept	2.24	0.25	**[1.79, 2.78]**
verb	1.46	0.34	**[0.83, 2.15]**
task	2.63	0.31	**[2.05, 3.29]**
verb:task	-0.10	0.44	[-0.95, 0.76]

Table 13: Summary of the Bayesian logistic regression model of LLaMA-3.3-70B model, Yes/No prompting.

Continuation prompting Like in binary choice prompting, Table 14 shows that the model only reveals the production-interpretation asymmetry, such that it chose more subject in interpretation than in production. There was no clear evidence for the IC verb type effect.

Formula: cont $\sim$ verb $*$ task + (1|itemID)			
	Estimate	Est. Error	95% CrI
Intercept	-3.66	0.66	**[-5.23, -2.68]**
verb	-0.61	0.88	[-2.67, 0.86]
task	6.83	1.27	**[4.89, 9.82]**
verb:task	1.10	1.74	[-1.78, 5.19]

Table 14: Summary of the Bayesian logistic regression model of LLaMA-3.3-70B model, continuation prompting.

Yes/No probability prompting Table 15 shows an interaction effect between verb and task type.

Formula: subject_yes_probability $\sim$ verb $*$ task $+$ (1|itemID)			
	Estimate	Est. Error	95% CrI
Intercept	0.73	0.01	**[0.70, 0.75]**
verb	0.12	0.02	**[0.07, 0.16]**
task	0.30	0.01	**[0.27, 0.33]**
verb:task	-0.05	0.03	**[-0.11, -0.00]**

Table 15: Summary of the Bayesian linear regression model of LLaMA-3.3-70B model, Yes/No probability prompting.

Pairwise comparisons in Table 16 and 17 further show that the verb type effect can be found in both production and interpretation task, and the asymmetry can be found in both IC1 and IC2 conditions.

Formula: subject_yes_probability $\sim$ verb/task $+$ (1|itemID)			
	Estimate	Est. Error	95% CrI
Intercept	0.73	0.01	**[0.71, 0.75]**
verb	0.12	0.02	**[0.07, 0.16]**
verbIC1:task	0.27	0.02	**[0.24, 0.31]**
verbIC2:task	0.33	0.02	**[0.29, 0.37]**

Table 16: Pairwise comparison of the verb type effect within the production and the interpretation task using Yes/No probability prompting in LLaMA-3.1-70B model.

Formula: subject_yes_probability $\sim$ task/verb $+$ (1|itemID)			
	Estimate	Est. Error	95% CrI
Intercept	0.73	0.01	**[0.70, 0.75]**
task	0.30	0.01	**[0.27, 0.33]**
taskProduction:verb	0.14	0.03	**[0.09, 0.19]**
taskInterpretation:verb	0.09	0.03	**[0.04, 0.14]**

Table 17: Pairwise comparison of the task type effect within IC1 and within IC2 verbs using Yes/No probability prompting in LLaMA-3.1-70B model.

B.3.3 QWen

Note that we only analyzed the responses of the continuation prompting in the interpretation task for `Qwen`, because models cannot be converged in other case. As can be seen below in Table 18, there is an opposite IC verb type effect such that the model referred to more subjects following IC2 verbs than IC1 verbs.

Formula: cont $\sim$ verb $+$ (1|itemID)			
	Estimate	Est. Error	95% CrI
Intercept	-0.60	0.07	**[-0.73, -0.47]**
verb	-0.37	0.13	**[-0.63, -0.11]**

Table 18: Summary of the Bayesian logistic regression model of `Qwen` model, continuation prompting.

B.3.4 GPT-4o

Recall that due to the extreme pattern of `GPT-4o` with binary promptings, no statistical model could be fitted. Below we only report results of statistical analyses for outputs from Yes/No, Yes/No probability, and continuation promptings only.

Table 19 shows the results of outcomes from Yes/No prompting of `GPT-4o`. As can be seen, there was a main effect of both verb type and task type, such that IC1 verbs and interpretation task elicited more choices towards subject antecedent. This aligns with the human performance.

Formula: yes_no $\sim$ verb $*$ task $+$ (1|itemID)			
	Estimate	Est. Error	95% CI
Intercept	-0.39	0.15	**[-0.69, -0.12]**
verb	0.68	0.29	**[0.15, 1.28]**
task	5.51	0.83	**[4.17, 7.42]**
verb:task	-0.23	0.46	[-1.15, 0.67]

Table 19: Summary of the Bayesian logistic regression model of `GPT-4o`, Yes/No prompting.

Table 20 shows the results of outcomes from Yes/No probability prompting of `GPT-4o`. Given statistical evidence for the interaction effect between verb and task type, we further ran a nested analysis for the effect of the task type within each level of verb, as shown in Table 11.

Formula: subject_yes_probability $\sim$ verb $*$ task $+$ (1|itemID)			
	Estimate	Est. Error	95% CI
Intercept	0.89	0.01	**[0.87, 0.90]**
verb	-0.01	0.01	[-0.04, 0.01]
task	-0.06	0.01	**[-0.08, -0.03]**
verb:task	0.13	0.02	**[0.08, 0.17]**

Table 20: Summary of the Bayesian linear regression model of `GPT-4o`, Yes/No probability prompting.

As can be seen, there was only a task effect limited to IC2 verbs and in an opposite way to human performance: the interpretation task was even less subject-biased than the production task,

as shown in the negative sign of the estimate of verbIC2:task.

Formula: subject_yes_probability $\sim$ verb/prompt $+ (1 \mid \text{itemID})$			
	Estimate	Est. Error	95% CI
Intercept	0.89	0.01	**[0.87, 0.90]**
verb	-0.01	0.01	[-0.04, 0.01]
verbIC1:task	0.01	0.02	[-0.03, 0.04]
verbIC2:task	-0.12	0.02	**[-0.15, -0.08]**

Table 21: Pairwise comparison of the task type effect within IC1 and within IC2 verbs using of GPT-4o, Yes/No probability prompting.

Lastly, Table 22 shows statistical results from continuation promptings of GPT-4o. The main effect of the task type shows that the model captures the production-interpretation asymmetry that the interpretation task was more subject-biased than the production task. However, the negative sign of the verb type effect shows a reversed IC verb effect, such that IC2 verbs were more subject-biased than IC1 verbs.

Formula: cont $\sim$ verb $*$ task $+ (1 \mid \text{itemID})$			
	Estimate	Est. Error	95% CI
Intercept	-1.43	0.30	**[-2.12, -0.95]**
verb	-3.11	0.63	**[-4.57, -2.10]**
task	5.85	1.04	**[4.26, 8.28]**
verb:task	0.10	0.60	[-1.05, 1.33]

Table 22: Summary of the Bayesian linear regression model of GPT-4o, continuation prompting.

Improving the Calibration of Confidence Scores in Text Generation Using the Output Distribution's Characteristics

Lorenzo Jaime Yu Flores[1,2] **Ori Ernst**[1,2] **Jackie Chi Kit Cheung**[1,2,3]

[1]Mila - Quebec AI Institute
[2]McGill University
[3]Canada CIFAR AI Chair
{lorenzo.flores, ori.ernst, cheungja}@mila.quebec

Abstract

Well-calibrated model confidence scores can improve the usefulness of text generation models. For example, users can be prompted to review predictions with low confidence scores, to prevent models from returning bad or potentially dangerous predictions. However, confidence metrics are not always well calibrated in text generation. One reason is that in generation, there can be many valid answers, which previous methods do not always account for. Hence, a confident model could distribute its output probability among multiple sequences because they are all valid. We propose task-agnostic confidence metrics suited to generation, which rely solely on the probabilities associated with the model outputs without the need for further fine-tuning or heuristics. Using these, we are able to improve the calibration of BART and Flan-T5 on summarization, translation, and QA datasets.

1 Introduction

Confidence scores are scores derived from a model's output, which are interpreted as the model's estimation of its own output's quality. These scores can be used in real-world applications to flag uncertain predictions in automated decision-making systems (Malinin and Gales, 2021), which could prompt further human review (Xiao et al., 2020), or force the model to abstain from answering when unsure (Liu et al., 2020; Kamath et al., 2020). To be useful, we want these scores to correlate with the output's quality.

A common approach to estimating confidence is through probability-based methods, which rely on the probabilities assigned by the model to output tokens. Most existing methods focus on the sequence with the highest probability, which we refer to as the top sequence (Murray and Chiang, 2018; Zablotskaia et al., 2023; Huang et al., 2023; Zhao et al., 2020; Perlitz et al., 2023; Malinin and Gales, 2021). A high probability for the top sequence suggests strong confidence in a particular prediction, while a lower value indicates uncertainty.

This approach is effective for tasks with a single correct answer. However, it faces significant challenges when applied to tasks with multiple valid outputs, as in many generation tasks. In such cases, a low top probability may not reflect a lack of confidence but rather that the model has identified several valid sequences (See Figure 1). Ideally, for open-ended tasks, a confident model would distribute high probabilities across multiple good sequences while assigning lower probabilities to less suitable options, while in classification, confidence can be indicated by a single high top probability.

To address this limitation, we propose new probability-based confidence estimation methods that consider the probabilities of multiple sequences instead of focusing solely on the top one. We introduce two methods: the first calculates the probability ratio between the highest-ranked sequences and the rest, while the second evaluates the thinness of the distribution's tail. Our experiments demonstrate that these metrics outperform existing baselines across three open-ended text generation tasks: translation, QA, and summarization.

2 Related Work

Probability-Based Methods These methods rely on the model outputs to compute token-level probabilities or entropy (Murray and Chiang, 2018; Zablotskaia et al., 2023; Zhao et al., 2020; Perlitz et al., 2023; Kumar and Sarawagi, 2019; Huang et al., 2023; Malinin and Gales, 2021). Other work uses natural language inference models to group similar sequences before computing entropy (Lin et al., 2023; Kuhn et al., 2023; Nikitin et al., 2024).

Similarity/Disagreement Based Methods When answers can be sampled from models (e.g., through dropout), self-consistency can be used

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 172–182
July 27 - August 1, 2025

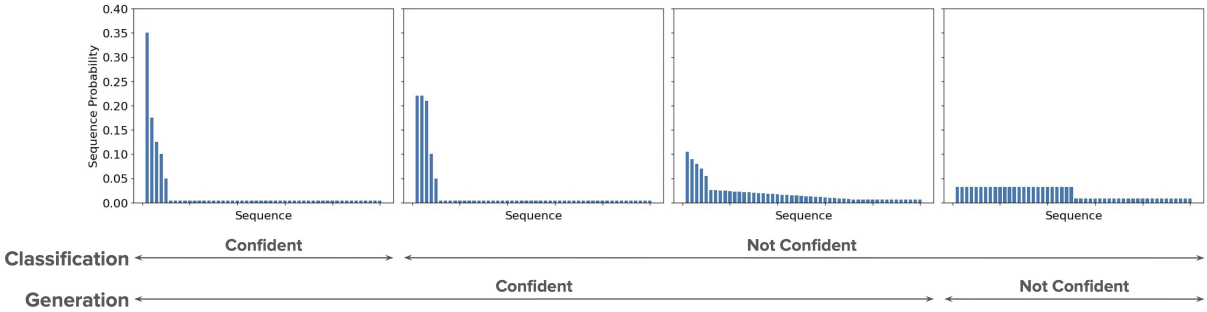


Figure 1: We illustrate the difference in interpretation of confidence in classification vs generation. Suppose a model generated output probability distributions for four different inputs; each bar is the prob. assigned to one class/sequence. In classification, only the 1st output would show model confidence, as it assigned most probability to one class. In generation, the first 3 outputs *could* show confidence because multiple sequences were valid.

to measure confidence: consistency across the top answers indicates confidence while variance indicates uncertainty (Xiao et al., 2020; Schmidt et al., 2022; Lakshminarayanan et al., 2017).

Fine-Tuning Based Methods In addition, other methods also fine-tune additional models to predict the correctness or confidence of the output (Yaldiz et al., 2024; Kamath et al., 2020; Malinin et al., 2019; Fathullah et al., 2023).

Out of Distribution Detection (OOD) Methods OOD can also be used to detect if a sample is in the training distribution, in which case a model is assumed to be confident (Liu et al., 2020; Vazhentsev et al., 2023).

Verbalized Confidence Scores With the increased conversational ability of LLMs, recent work prompts the model to give a confidence score with its answer (Lin et al., 2022; Tian et al., 2023; Kapoor et al., 2024; Han et al., 2024).

Our work is closest to the probability-based methods; they are easily adaptable and task agnostic. They do not require metrics or NLI models to measure similarity, computation for OOD detection, or models that can verbalize their confidence.

3 Method

Problem Definition We define confidence as a score computed using the model outputs, that describes its assessment of its prediction quality. We want to compute the model's sample-level confidence for its output, that is positively correlated (i.e. calibrated) to the output's quality, measured by the evaluation metric used for the task (e.g. automated metrics, human evaluation). Formally,

$$\text{Confidence}(x, \hat{y}, \phi) \propto \text{Quality}(y, \hat{y}),$$

where x is the input, y is the target, $\hat{y}$ is the prediction, and ϕ are the model parameters.

At inference time, we run beam search to generate N sequences. Each sequence's probability is obtained by taking the product of the individual token probabilities. Given the i-th beam $\hat{y}^{(i)}$:

$$p_{\hat{y}^{(i)}}(x) = \prod_t p(\hat{y}_t^{(i)} | \hat{y}_{<t}^{(i)}, x)$$

Methods We account for the fact that there can be multiple valid outputs by measuring two characteristics that we hypothesize are present in all confident outputs regardless of the number of valid sequences (See Figure 2). The first characteristic of a confident model is that it distinguishes good from average or bad sequences, and subsequently assigns higher probability to a select set of sequences it deems as good compared to other sequences.

Ratio This motivates the ratio method: we measure how much more confident the model is in one of its *best* beams $p_{\hat{y}^{(1)}}$, versus one of its *average* beams $p_{\hat{y}^{(k)}}$, where $p_{\hat{y}^{(1)}}$ to $p_{\hat{y}^{(k)}}$ are sorted in descending order. This captures the intuition that a confident model will assign more probability to its best sequence than to an average sequence, whereas an unconfident model would assign similar probabilities to them. We expect the optimal value of k to differ by task, as different tasks can have different levels of diversity in valid generation outputs. Hence, we tune k on a validation set, and report its performance on the test set in the results.

$$\text{Ratio}(x) = \frac{p_{\hat{y}^{(1)}}(x)}{p_{\hat{y}^{(k)}}(x)}$$

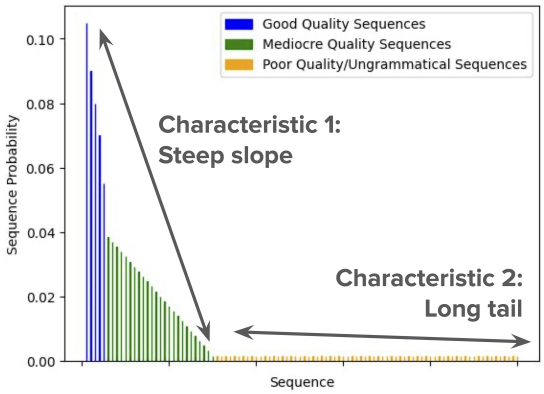


Figure 2: We hypothesize that a confident model's output *would* have a steep slope and long tail; colors added for illustration purposes only.

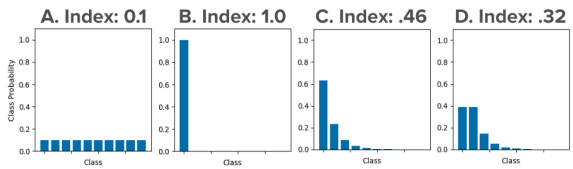


Figure 3: Samples of distributions and their tail indices

The second characteristic of a confident model is that it will assign low probability to many *bad* sequences. Suppose each sequence is a "class" – in Figure 3B, 3C, and 3D, the right-most classes have very small probabilities, and hence have a "thin tail"; all three figures exhibit this property, regardless of how many *correct* sequences they have. In contrast, an unconfident output where all classes receive equal probability (Figure 3A) has a thick tail. We quantify this with the tail index.

Tail Thinness We adapt the tail index proposed by Huang (2024), originally designed to measure the thinness of statistical distributions. The higher the tail thinness, the thinner the tail.

$$\text{Tail Thinness}(x) = \sum_{i=1}^{N} p_{\hat{y}^{(i)}}(x)^2$$

This sums the squared sequence probabilities for all N sequences generated using beam search. Because the probabilities for N sequences do not sum to 1, we first normalize them using softmax. We report the temperature used in Appendix A. Applying this to Figure 3, the uniform distribution (Fig A) gets a small tail thinness, while a degenerate distribution (Fig B) has the highest tail thinness. The metric also assigns similar scores to distributions with similar tail thicknesses (Figs C and D). While this performs similarly to sequence-level entropy (Appendix C), we use tail-thinness metric (Huang, 2024) as it better describes the shape of the tail for which we have formulated our assumptions.

4 Experiments

Fine-tuning and Inference We first perform supervised fine-tuning (SFT) with BART Base (Lewis et al., 2019) or Flan-T5 Base (Chung et al., 2022), both relatively small models with no prior ability to verbalize confidence (Appendix A). After SFT, we generate the confidence scores for the test set. We get the sequence probabilities the top $N = 100$ sequences, using beam search from HuggingFace (Wolf et al., 2020), and replicate the baselines.

Evaluation We compute the Spearman correlation between the confidence scores and the quality scores, similar to analyses by Zablotskaia et al. (2023); Malinin and Gales (2021). We evaluate the top beam against the reference using ROUGE-L (Lin, 2004) for summarization, BLEU (Papineni et al., 2002) for translation, or F1 for question answering, and test for statistical significance with a bootstrap test (Berg-Kirkpatrick et al., 2012).

Baselines We report the equations that we replicate from previous literature, shown in Table 1.

For the probability based methods (Rows 1-4), we compute (1) ATP: average token probability for the top sequence (Murray and Chiang, 2018; Zablotskaia et al., 2023), (2) ATE: average token entropy for the top sequence (Zhao et al., 2020; Perlitz et al., 2023), (3) DAE: dropout-based average token entropy across 10 outputs (Eq 1) (Malinin and Gales, 2021), and (4) WTP: weighted average of the top-K sequences' average token log probabilities (Eq 2) (Malinin and Gales, 2021).

For the similarity/disagreement based methods (Rows 5-7), we use dropout and sample 10 outputs per instance. We compute the (1) DSM: dropout similarity using METEOR (Eq 3) (Schmidt et al., 2022), (2) DVB: dropout variance using BLEU (Eq 4) (Xiao et al., 2020), and (3) DVK: dropout variance between token probabilities using KL divergence (Eq 5) (Lakshminarayanan et al., 2017).

$$\text{Conf}_{\text{DAE}} = \frac{1}{10}\sum_{i=1}^{10}\frac{1}{|\hat{y}^{(i)}|}\sum_{t=1}^{|\hat{y}^{(i)}|}\mathcal{H}\left(p(\hat{y}_t^{(i)}|\hat{y}_{<t}^{(i)}, x)\right) \quad (1)$$

$$\mathcal{H}\left(p(\hat{y}_t^{(i)}|\hat{y}_{<t}^{(i)}, x)\right) = -\sum_{j=1}^{|\mathcal{V}|} p(\hat{y}_{t,j}^{(i)}|\hat{y}_{<t}^{(i)}, x)\log\left(p(\hat{y}_{t,j}^{(i)}|\hat{y}_{<t}^{(i)}, x)\right)$$

$$\text{Conf}_{\text{WTP}} = -\sum_{i=1}^{10} \pi_i \left(\frac{1}{|\hat{y}^{(i)}|} \ln(p(\hat{y}^{(i)})) \right) \quad (2)$$

$$\pi_i = \frac{\exp\left(\frac{1}{|\hat{y}^{(i)}|} \ln(p(\hat{y}^{(i)})) \right)}{\sum_{j=1}^{10} \exp\left(\frac{1}{|\hat{y}^{(j)}|} \ln(p(\hat{y}^{(j)})) \right)}$$

$$\ln(p(\hat{y}^{(i)})) = \sum_{t=1}^{|\hat{y}^{(i)}|} \ln(p(\hat{y}_t^{(i)} | \hat{y}_{<t}^{(i)}, x))$$

$$\text{Conf}_{\text{DSM}} = \frac{\sum_{i=1}^{10} \sum_{j=1}^{10} \text{Meteor}(\hat{y}^{(i)}, \hat{y}^{(j)})}{N(N-1)} \quad (3)$$

$$\text{Conf}_{\text{DVB}} = \sum_{i=1}^{10} \sum_{j=1}^{10} (1 - \text{BLEU}(\hat{y}^{(i)}, \hat{y}^{(j)}))^2 \quad (4)$$

$$\text{Conf}_{\text{DVK}} = \sum_{i=1}^{10} KL(p(\hat{y}^{(i)}|x), p_{\bar{y}}) \quad (5)$$

$$\bar{y}_{\text{Prob}} = \frac{1}{10} \sum_{i=1}^{10} p(\hat{y}^{(i)}|x)$$

Where $\hat{y}^{(i)}$ is the decoded sequence i sampled by activating dropout, $\hat{y}_t^{(i)}$ is the t-th output token for sequence i, and $\hat{y}_{t,j}^{(i)}$ is the j-th vocabulary at position t for sequence i.

Datasets We test on **Translation** (1) WMT 2017 English-German (Bojar et al., 2017), (2) WMT 2017 English-Russian (Bojar et al., 2017), (3) FLORES (Filipino Set) (NLLB, 2022), **Question Answering** (1) SQUAD (Rajpurkar et al., 2016), (2) HotpotQA (Yang et al., 2018), **Summarization** (1) DebateSumm (Roush and Balaji, 2020), (2) Reddit-TiFu (Kim et al., 2018), (3) XSUM (Narayan et al., 2018), (4) CNN-DailyMail (See et al., 2017)

5 Results

We report the correlation between the evaluation metric and confidence scores in Table 1 (See Appendix A for details). For BART, our methods achieve better correlation on 6 out of 9 datasets. We see larger gains in translation and question answering, as compared to summarization. The tail thinness method generally yields larger improvements (up to +17.2%) than the ratio method (up to +16.1%). For Flan-T5, our methods also achieve better correlation on 4 out of 9 datasets. Like for BART, we observe larger improvements using the tail thinness (up to +10.0%) method than the ratio based method (up to +8.3%). Overall, our methods yield the best performance more frequently than previous methods across all dataset-model pairs (tail thinness: 10/16, ratio: 8/16, DSM: 4/16), with median rankings of 2 and 3 for the tail and ratio methods (next being ATP, rank 4).

Robustness to Multiple Valid Sequences Qualitatively, we find that our methods assign high confidence to outputs where there are multiple valid sequences. We look at examples where our metrics assigned high confidence, but other methods like average token log probability assigned low confidence (See Figure 4). In these examples, there were indeed multiple, correct outputs; this resulted in lower probability for the top beam (2nd and 3rd image). If we only used the top beam's probability to measure confidence, we might conclude that the model is unconfident. In contrast, our methods which rely on the ratio of sequence probabilities and tail thinness, rather than the top probability, are able to correctly identify that the model is still confident in such scenarios. This illustrates how using features of the distribution like slope or tail thinness can be more indicative of confidence in text generation, rather than solely looking at the features of the top output.

Failure Cases We examine samples for which the confidence scores are not well calibrated. Looking at the FLORES (Filipino) for Flan-T5, we observed samples where the model was confident, but its output was bad. Here, the model failed to translate a few key terms, which changed the meaning of the sentence (See Table 6). Other times, the confidence scores were well calibrated, but the quality score was not estimated well. This stemmed from noisy labels or limitations of the evaluation metric (See Table 7) which may require future work.

Choice of k In general, open ended tasks (translation, summarization) benefited from larger values for k, and close-ended tasks (QA) from smaller values of k (See Figure 5). One explanation for this could be that k serves as a parameter which delineates the *good* vs. *average* sequences. Finding the k that best separates the two groups allows us to most accurately measure the difference in confidence between both groups. Open-ended tasks can have more good sequences, hence correlation is maximized when we choose a higher value for k. In contrast, close-ended tasks have fewer good sequences, so a lower value for k is better.

We note that the optimal value for k may be very large for open-ended tasks like summarization. In our experiments, we capped k to 100 due to computational limitations, which may have underestimated its optimal value, and thus explain why our methods are less competitive than the baselines.

		Fil–EN		DE–EN		RU–EN		HotpotQA		SQUAD		Debate		Reddit		CNN		XSUM		Rank	
		Bt	FT5	Bt	FT5	Bt	FT5	Bt	FT5	Bt	FT5	Bt	FT5	Bt	FT5	Bt	FT5	Bt	FT5	Avg	Med
Probability	ATP	.473	.468	.028	.370	.530	.023	.209	.302	.391	.577	.447	.247	**.618**	**.577**	.109	.156	.119	.078	4.4	4
	ATE	.308	.335	.035	.297	.437	.042	.051	.152	.094	.049	.416	.248	.615	.474	.020	.138	.093	.082	6.4	7
	DAE	.217	.161	.346	.294	.230	.178	.242	.367	.327	.226	.135	.037	.049	.049	**.295**	**.380**	.314	.353	5.4	6
	WTP	.516	.495	.162	.287	.602	.055	.130	.180	.179	.020	.489	.253	.616	.575	.106	.162	.120	.063	5	5
Sim/Diff	DSM	.441	**.508**	.424	**.462**	.374	.486	.168	.270	.394	.332	.192	.038	.038	.167	.255	.323	**.323**	**.383**	4.4	4.5
	DVB	.455	.489	.512	.461	.409	.488	.043	.000	.378	.467	.144	.061	.058	.143	.264	.325	.305	.363	4.7	4.5
	DVK	.001	.008	.110	.064	.110	.013	.177	.232	.340	.426	.063	.025	.045	.059	.065	.070	.103	.117	7.6	8
Ours	Ratio	**.546**	.200	★**.653**	.209	★**.768**	**.491**	**.249**	.360	★**.505**	.565	**.496**	☆**.293**	.596	.304	.103	.055	.082	.196	3.9	3
	Tail	★**.649**	.380	★**.648**	.190	★**.779**	**.506**	**.255**	★**.451**	★**.493**	**.582**	**.518**	★**.354**	.601	.300	.100	.031	.131	.212	3.2	2

Table 1: Spearman correlation (absolute value) between confidence score and evaluation metric/quality score (BLEU for translation, F1 for QA, RougeL for summarization); Bt: BART, FT5: Flan-T5, stars indicate significant difference from next best method (bootstrap test, ☆$\alpha = 0.10$, ★$\alpha = 0.05$)

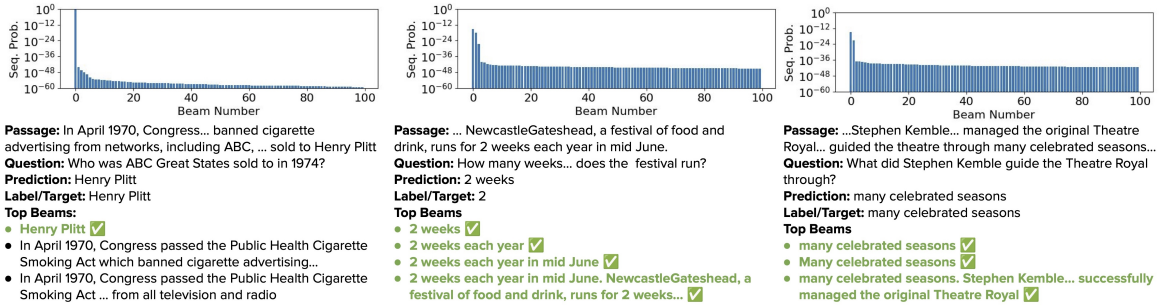


Figure 4: Samples from SQUAD (Rajpurkar et al., 2016); 1st image only has one valid output, whereas the 2nd and 3rd have multiple; our tail-thinness and ratio based confidence correctly assign high confidence to all samples, but avg. log prob. only assigns high confidence to the first image (Note: The y-axis is plotted on the log scale)

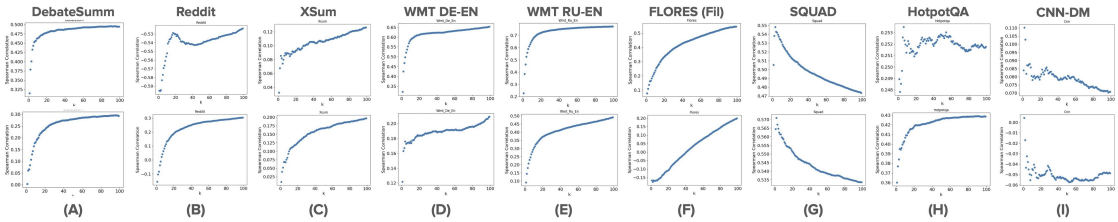


Figure 5: Spearman Correlation vs k on test set for BART (top row) and Flan-T5 (bottom row); In general, open-ended tasks (summarization: A-C, translation: D-F) benefit from larger k, close-ended tasks (QA: G-H, Reddit: I) use smaller k

6 Conclusion

We identified characteristics of output distributions from a confident model in generation tasks, and used this to propose metrics that capture these characteristics. We find that on various datasets, these characteristics are better correlated to quality metrics than previous methods.

There are various directions future work can take to improve the computation and evaluation of confidence scores. For example, we observed that models can be overly confident in wrong answers (Table 6), which can lead to miscalibration when using our methods. Future work can study the reasons for model overconfidence, and propose ways to reduce or account for this. In addition, more work is required to improve the evaluation of calibration in generation tasks. Concretely, calibration is well defined for classification tasks when there is a binary outcome, and metrics like expected calibration error (Tian et al., 2023) measure whether the model's confidence scores match its accuracy (e.g. A model should be right in 90% of the examples for which it claimed to be 90% confident). More work can be done to develop metrics that capture a similar notion of calibration in generation tasks where the outcome is continuous.

Limitations and Potential Risks

One limitation is that our methods require users to compute k beams, both when tuning hyperparameters, and at inference time. This can be computationally expensive, especially for open-ended tasks where the optimal value of k may be large. Hence, future work may find more efficient ways of estimating the confidence.

Another caveat is that we only evaluate the performance of our methods on a limited set of models and parameters. For example, we fine-tuned various models with early stopping. Hence, future work which seeks to use these methods must re-evaluate the scores on their tasks, to avoid deploying miscalibrated confidence scores in practical settings.

Finally, future work could study better ways to evaluate confidence scores; we found that traditional evaluation metrics may lead to poor quality ratings, and it was difficult to find datasets with human evaluation scores to use.

Acknowledgements

We would like to thank Ines Arous, Ziling Cheng, Zichao Li, and Caleb Moses for providing useful writing feedback, and David Austin, Cesare Spinoso-di Piano, and Xiyuan Zou for engaging in productive discussions as we developed the paper. We also thank NVIDIA for their support in terms of computational resources. This work was also supported in part by the IVADO Postdoctoral Fellowship.

References

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea. Association for Computational Linguistics.

Ond rej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Yassir Fathullah, Guoxuan Xia, and Mark John Francis Gales. 2023. Logit-based ensemble distribution distillation for robust autoregressive sequence uncertainties. *ArXiv*, abs/2305.10384.

Haixia Han, Tingyun Li, Shisong Chen, Jie Shi, Chengyu Du, Yanghua Xiao, Jiaqing Liang, and Xin Lin. 2024. Enhancing confidence expression in large language models through learning from past experience.

Hening Huang. 2024. A new measure of the tail-heaviness of a probability distribution.

Yuheng Huang, Jiayang Song, Zhijie Wang, Shengming Zhao, Huaming Chen, Felix Juefei-Xu, and Lei Ma. 2023. Look before you leap: An exploratory study of uncertainty measurement for large language models.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. 2024. Large language models must be taught to know what they don't know.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2018. Abstractive summarization of reddit posts with multi-level memory networks.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *ArXiv*, abs/1903.00802.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training

for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *Trans. Mach. Learn. Res.*, 2024.

Jeremiah Zhe Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. 2020. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction.

Andrey Malinin, Bruno Mlodozeniec, and Mark John Francis Gales. 2019. Ensemble distribution distillation. *ArXiv*, abs/1905.00076.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Brussels, Belgium. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don‘t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *ArXiv*, abs/2405.20003.

Team NLLB. 2022. No language left behind: Scaling human-centered machine translation.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Yotam Perlitz, Ariel Gera, Michal Shmueli-Scheuer, Dafna Sheinwald, Noam Slonim, and Liat Ein-Dor. 2023. Active learning for natural language generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9862–9877, Singapore. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Allen Roush and Arvind Balaji. 2020. DebateSum: A large-scale argument mining and summarization dataset. In *Proceedings of the 7th Workshop on Argument Mining*, pages 1–7, Online. Association for Computational Linguistics.

Maximilian Schmidt, A. Bartezzaghi, Jasmina Bogojeska, Adelmo Cristiano Innocenza Malossi, and Thang Vu. 2022. Combining data generation and active learning for low-resource question answering. In *International Conference on Artificial Neural Networks*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Artem Vazhentsev, Akim Tsvigun, Roman Vashurin, Sergey Petrakov, Daniil Vasilev, Maxim Panov, Alexander Panchenko, and Artem Shelmanov. 2023. Efficient out-of-domain detection for sequence to sequence models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1430–1454, Toronto, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Tim Z. Xiao, Aidan N. Gomez, and Yarin Gal. 2020. Wat zei je? detecting out-of-distribution translations with variational transformers.

Duygu Nur Yaldiz, Yavuz Faruk Bakman, Baturalp Buyukates, Chenyang Tao, Anil Ramakrishna, Dimitrios Dimitriadis, and Amir Salman Avestimehr. 2024. Do not design, learn: A trainable scoring

function for uncertainty estimation in generative llms. *ArXiv*, abs/2406.11278.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Polina Zablotskaia, Du Phan, Joshua Maynez, Shashi Narayan, Jie Ren, and Jeremiah Liu. 2023. On uncertainty calibration and selective generation in probabilistic neural summarization: A benchmark study. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2980–2992, Singapore. Association for Computational Linguistics.

Yuekai Zhao, Haoran Zhang, Shuchang Zhou, and Zhihua Zhang. 2020. Active learning approaches to enhancing neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1796–1806, Online. Association for Computational Linguistics.

A Implementation Details

All models were fine-tuned on one NVIDIA A100 GPU, with a constant learning rate 5e-5, and batch size of 10. The scripts and fine-tuned models are provided in the repository. Roughly 80 hours were used to train and perform inference on one GPU.

During SFT, we train for at most 3 epochs. We observe overfitting on many datasets, and remedy this by employing early stopping, where we stop training if the loss on the validation set does not improve after 2 steps. This was applied to all datasets except HotpotQA, WMT RU-EN, and DebateSumm. We report the number of SFT steps in Table 2.

Dataset	BART	Flan-T5
WMT DE-EN	200	200
WMT RU-EN	6000	6000
FLORES Filipino	260	200
SQUAD	220	240
HotpotQA	26835	26835
DebateSumm	1500	1500
Reddit	140	200
CNN	200	200
XSUM	120	200

Table 2: Number of Fine-Tuning Steps Taken per Task and Model

We report the parameters used for the ratio and tail-thinness methods (k: ratio method, temperature: softmax for the tail method) in Table 3.

Dataset	Model	k	Temp
FLORES Filipino	BART	99	1.000
	Flan-T5	99	1.000
WMT DE-EN	BART	99	1.000
	Flan-T5	99	1.000
WMT RU-EN	BART	79	1.000
	Flan-T5	99	1.000
HotpotQA	BART	1	0.010
	Flan-T5	1	0.050
SQUAD	BART	1	0.050
	Flan-T5	4	0.001
DebateSumm	BART	95	1.000
	Flan-T5	85	1.000
Reddit	BART	2	0.005
	Flan-T5	99	0.010
CNN	BART	3	0.001
	Flan-T5	77	0.001
XSUM	BART	4	0.100
	Flan-T5	98	0.100

Table 3: Fine-Tuning Parameters for Various Tasks

B Dataset Details

Licenses The FLORES, SQUAD, and HotpotQA datasets were used under the Creative Commons Attribution Share Alike 4.0 license; DebateSumm, XSUM, and Reddit-TiFu used the MIT license, the CNN DailyMail dataset used Apache2.0, and WMT17 did not provide a license on the HuggingFace platform.

Data Splits For training and inference efficiency, we only use subsets of the datasets in some cases. The scripts used to generate the datasets are provided in the repository. At a high level, we take and shuffle the original dataset, then generate a train and test split from that. We perform inference on the test set, for which we report the statistics in the results section. Note that because we employ early stopping, the full training set is not necessarily provided. The number of steps actually taken are reported in Appendix A.

C Analysis of Additional Baselines

In addition to the baselines from previous literature, we compare our method to two baselines that can

Dataset	Train	Val	Test
FLORES Filipino	900	97	1012
WMT DE-EN	2000	100	1000
WMT RU-EN	20000	100	1000
HotpotQA	89447	100	1000
SQUAD	87599	100	1000
DebateSumm	5000	100	1000
Reddit	2000	100	1000
CNN	2000	100	1000
XSUM	20000	100	1000

Table 4: Data Splits by Task

be viewed as straightforward methods of creating a confidence score, namely beam-level entropy (Eq 6) and the sum of the top-k beam probabilities (Eq 7), and report the results in Table 5.

Overall, we find that the performance of the tail method is similar to that of beam-level entropy, although the original tail method performs slightly better in multiple cases.

As for the Top-K baseline, both our ratio and tail methods outperform it on translation and QA tasks for BART, and on QA, English-Russian, DebateSumm, and XSum for T5. However, it underperforms top-K on other tasks, particularly in summarization, which is similar to our findings for the WTP method presented in the main paper.

$$p(\hat{y}^{(i)}) = \sum_{t=1}^{|\hat{y}^{(i)}|} p(\hat{y}_t^{(i)} | \hat{y}_{<t}^{(i)}, x)$$

$$p_{\text{softmax}}(\hat{y}^{(i)}) = \frac{\exp(p(\hat{y}^{(i)}))}{\sum_{j=1}^{k} \exp(p(\hat{y}^{(j)}))}$$

$$\text{Conf}_{\text{Beam Entropy}} = -\sum_{i=1}^{k} p_{\text{softmax}}(\hat{y}^{(i)}) \ln(p_{\text{softmax}}(\hat{y}^{(i)})) \quad (6)$$

$$\text{Conf}_{\text{Top K Probs}} = \sum_{i=1}^{k} p(\hat{y}^{(i)}) \quad (7)$$

D Failure Case Examples

We provide examples of cases where there is miscalibration, either due to actual model miscalibration (Table 6), or due to issues with the evaluation strategy (Table 7).

		Fil–EN		DE–EN		RU–EN		HotpotQA		SQUAD		Debate		Reddit		CNN		XSUM	
		Bt	FT5	Bt	FT5	Bt	FT5	Bt	FT5	Bt	FT5	Bt	FT5	Bt	FT5	Bt	FT5	Bt	FT5
Ours	Tail	0.649	0.380	0.648	0.190	0.779	0.506	0.255	0.451	0.494	0.582	0.518	0.354	0.601	0.300	0.100	0.031	0.131	0.212
	Ratio	0.546	0.200	0.653	0.209	0.768	0.491	0.249	0.360	0.505	0.565	0.496	0.293	0.596	0.304	0.103	0.055	0.082	0.196
Base	Entropy	0.649	0.381	0.648	0.190	0.779	0.507	0.256	0.448	0.472	0.565	0.518	0.354	0.594	0.346	0.101	0.043	0.135	0.215
	Sum Top K	0.516	0.495	0.165	0.285	0.603	0.058	0.103	0.144	0.133	0.006	0.490	0.253	0.616	0.575	0.106	0.162	0.120	0.063

Table 5: Spearman correlation (absolute value) between confidence score and evaluation metric/quality score (BLEU for translation, F1 for QA, RougeL for summarization); Bt: BART, FT5: Flan-T5

Overconfident Model: Wrong Translation
Source: Translate English to Filipino: In the archipelagos and lakes you do not necessarily need a yacht
Prediction: Ang mga archipelago at mga lupa ay hindi nangangailangan ng isang yacht. (*Archipelagos and **land** do not need a **yacht***)
Target: Sa mga arkipelago at mga lawa ay hindi mo naman palaging kakailanganin ang yate. (*In archipelagos and lakes, you do not always need a yacht.*)
Overconfident Model: Wrong Translation
Source: Translate English to Filipino: Scotturb Bus 403 travels regularly to Sintra, stopping at Cabo da Roca
Prediction: Ang Scotturb Bus 403 ay nagsimula sa Sintra, na nagsimula sa Cabo da Roca. (*The Scotturb Bus 403 **starts** from Sintra, and **starts** from Cabo de Roca*)
Target: Regular na bumibiyahe ang Scotturb Bus 403 patungong Sintra, tumitigil sa Cabo da Roca. (*The Scotturb Bus 403 regularly travels to Sintra, stopping at Cabo da Roca*)

Table 6: Examples of outputs where the confidence scores themselves are miscalibrated, taken from the FLORES (Filipino) Dataset (NLLB, 2022)

Good Output Rated as Bad: Correct Gist, Different Style
Source: Manchester United winger Ashley Young celebrated his prominent role in his side's derby win by laughing at 'noisy neighbours' City. The 29-year-old, who scored United's first goal, and created two more for Marouane Fellaini and Chris Smalling, said Louis van Gaal's side were focused on quieting the Manchester City fans... 'Confidence has been sky high for a long time and we want to keep the momentum going now. ... I think it was over when it went to 3-1
Prediction: Ashley Young celebrated his prominent role in his side's derby win by laughing at 'noisy neighbours' City fans. The 29-year-old, who scored United's first goal, and created two more for Marouane Fellaini and Chris Smalling, said Louis van Gaal's side were focused on quieting the Manchester City fans.
Target: Ashley Young scored Manchester United equaliser in 4-2 win. 'We quietened them down straight away' claims Young, as he praises fans. Young claims the game was over once United went 3-1 ahead. England winger says confidence at Old Trafford is now 'sky high'. CLICK HERE to read Ian Ladyman's match report from Old Trafford. READ: Man Utd runaway league leaders in table vs the current top seven
Good Output Rated as Bad: Noisy Label
Source: My first trip to New York was with my aunt when I was about 18 and I had been appearing in Coronation Street for about 12 months... we went to see the musical Chicago on Broadway and did lots and lots of shopping... one of the first things my boyfriend Sylvain Longchambon and I did on my latest trip was to visit the site of the World Trade Center to see the monument to those who died... Coronation Street star Samia Ghadie and boyfriend Sylvain Longchambon take a selfie in New York... The JW Marriott Essex House as seen from Central Park, of which it boasts panoramic views... On the second night of our break, we went on the Bateaux New York cruise, which offers an unforgettable three-hour trip along the Hudson River... The couple, who met as partners on Dancing on Ice, enjoyed a Bateaux New York Cruise of the Hudson...We were wondering who was going to emerge when out stepped Lady Gaga. It was the ultimate holiday treat - it was just a shame I couldn't arrange a selfie with her. That would have been the icing on the cake. American Sky (0843 636 2001) offers a three-night holiday to New York from £899pp. This includes accommodation in the JW Marriott Essex House New York on a room-only basis and return flights with British Airways from Heathrow. Visit nycgo com for more information
Prediction: My first trip to New York was with my aunt when I was about 18 and I had been appearing in Coronation Street for about 12 months. We had a lot of fun together - we went to see the musical Chicago on Broadway and did lots and lots of shopping. We returned to the city that never sleeps in January 2001, eight months before the terrible events of 9/11, and one of the first things my boyfriend Sylvain Longchambon and I did on my latest trip was to visit the site of the World Trade Center to see the monument to those who died.
Target: Samia Ghadie and boyfriend Sylvain Longchambon spent four days in NYC. While there the Dancing on Ice stars stayed at the JW Marriott Essex House. Their suite on the 31st floor offered up panoramic views of Central Park. The couple enjoyed a three-hour Bateaux New York cruise on the Hudson. The confirm Burger Joint in Le Parker Meridien Hotel lives up to the hype.
Bad Output Rated as Good: Repeated Phrase
Source: SOURCE: An exotic animal owner from Ohio is campaigning for his bear and big cats to be returned after they were seized by the state in January. In total, six tigers, a bear, a lion, a cougar, a black leopard and a liger (part lion, part tiger) were taken from Kenny Hetrick's Stony Ridge farm after it was found he did not have the correct permit and cages were 'unsafe'. Authorities have also started cracking down on the owners of wild creatures following an incident in 2011 where a man in eastern Ohio released 56 exotic animals - including lions and tigers - then killed himself. Apparently Hetrick has been heartbroken by the clampdown on his farm and the 72-year-old widower is now fighting to overturn the seizure, ... 'He's lost without them,' said Josh Large, who lives four houses away...
Prediction: In total, six tigers, a bear, a lion, a cougar, a black leopard and a liger were taken from Kenny Hetrick's Stony Ridge farm. In total, six tigers, a bear, a lion, a cougar, a black leopard and a liger were taken from Kenny Hetrick's Stony Ridge farm after it was found he did not have the correct permit and cages were 'unsafe'. Authorities have also started cracking down on the owners of wild creatures following an incident in 2011 where a man in eastern Ohio released 56 exotic animals - including lions and tigers - then killed himself. Hetrick has been heartbroken by the clampdown on his farm and the 72-year-old widower is now fighting to overturn the seizure, backed by neighbors who insist his menagerie doesn't pose a threat.
Target: In total, six tigers, a bear, a lion, a cougar, a black leopard and a liger (part lion, part tiger) were taken from Kenny Hetrick's Stony Ridge farm . State officials found he didn't have the right permit and cages were 'unsafe' But now the 72-year-old is fighting to overturn the seizure, backed by neighbors who insist his menagerie doesn't pose a threat . 'He's lost without them,' said Josh Large, who lives four houses away .

Table 7: Examples of outputs where the outputs are rated incorrectly based on the metric, taken from the CNN-DailyMail Dataset (See et al., 2017)

KNOWSHIFTQA: How Robust are RAG Systems when Textbook Knowledge Shifts in K-12 Education?

Tianshi Zheng*♠, Weihan Li*♣, Jiaxin Bai♠, Weiqi Wang♠, Yangqiu Song♠
♠Department of Computer Science and Engineering, HKUST, Hong Kong SAR, China
♣Department of Mechanical Engineering, The University of Tokyo, Tokyo, Japan
tzhengad@connect.ust.hk

Abstract

Retrieval-Augmented Generation (RAG) systems show remarkable potential as question answering tools in the K-12 Education domain, where knowledge is typically queried within the restricted scope of authoritative textbooks. However, discrepancies between these textbooks and the parametric knowledge inherent in Large Language Models (LLMs) can undermine the effectiveness of RAG systems. To systematically investigate RAG system robustness against such knowledge discrepancies, we introduce **KNOWSHIFTQA**[1]. This novel question answering dataset simulates these discrepancies by applying deliberate hypothetical knowledge updates to both answers and source documents, reflecting how textbook knowledge can shift. KNOWSHIFTQA comprises 3,005 questions across five subjects, designed with a comprehensive question typology focusing on context utilization and knowledge integration. Our extensive experiments on retrieval and question answering performance reveal that most RAG systems suffer a substantial performance drop when faced with these knowledge discrepancies. Furthermore, questions requiring the integration of contextual (textbook) knowledge with parametric (LLM) knowledge pose a significant challenge to current LLMs.

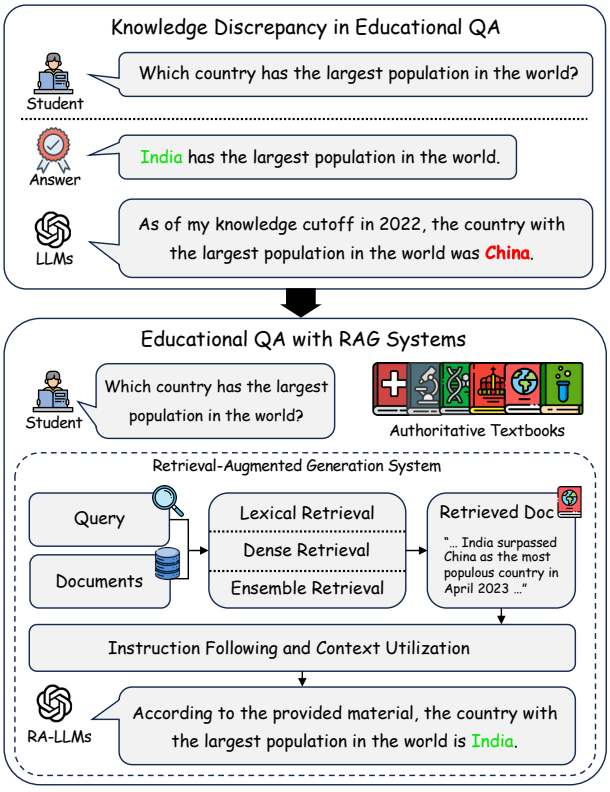


Figure 1: An illustration of knowledge discrepancy in educational QA and the application of RAG systems.

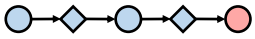
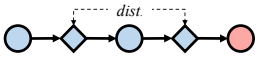
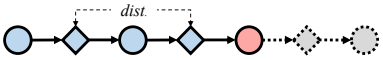
1 Introduction

In K-12 education, Question Answering (QA) systems serve as an important resource for learning assistance, where answers are precisely provided within a restricted knowledge scope from authoritative sources (i.e., textbooks) (Raamadhurai et al., 2019; Soares et al., 2021). Meanwhile, benefiting from the emergent abilities (Wei et al., 2022) of Large Language Models (LLMs) and advanced information retrieval (IR) methods, Retrieval-Augmented Generation (RAG) systems have achieved remarkable performance in various knowledge-intensive tasks in natural language processing (Lewis et al., 2020; Jiang et al., 2023; Gao et al., 2024), demonstrating their great potential as QA systems in K-12 education (Gan et al., 2023; Kasneci et al., 2023; Yan et al., 2024).

In K-12 educational QA, one of the primary concerns is ensuring that the knowledge conveyed in the answer is consistent with the officially designated textbooks (Extance, 2023). However, notable knowledge shifts (i.e. factuality discrepancies) may exist between the knowledge in textbooks and the internal knowledge of LLMs, due to the evolving nature of facts (Arbesman, 2012), updates in peda-

* Equal Contribution
[1] https://github.com/HKUST-KnowComp/KnowShiftQA

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 183–195
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

Question Type	Reasoning Pattern and Example Question	Hypothetical Knowledge Update
Simple Direct		NV goggles - detect - ~~Infrared~~ light ***Ultraviolet***
	What type of light is *detected* by **night vision goggles**?	Original: Infrared Light Updated: Ultraviolet Light
Multi-hop Direct		de Broglie eq. - developed by - ~~Louis~~ de Broglie ***Maurice***
	Which scientist *developed* an **equation** that can *calculate* the **wavelength of a particle**?	Original: Louis de Broglie Updated: Maurice de Broglie
Multi-hop Distant		~~Na^+/K^+~~ Pump - creates - EC Gradient ***Ca^{2+}***
	Which pump *creates* an **electrochemical gradient** that *enables* **secondary active transport** to occur?	Original: Sodium-potassium Pump Updated: Calcium Pump
Multi-hop Implicit		~~Polonium~~ - found in - Uranium ores ***Thorium***
	Who *discovered* the **radioactive element** that is commonly *found* in **uranium ores**?	Original: Marie Curie Updated: Jöns Jacob Berzelius
Distant Implicit		~~Mitochondrion~~ - conducts - Cellular respiration ***Golgi apparatus***
	Who *discovered* the **organelle** that is *responsible* for the **biological process** that *produces* **ATP**?	Original: Albert von Kölliker Updated: Camillo Golgi

subject/object · predicate · contextual fact · updated fact · parametric fact · *dist.* distant fact

Table 1: Five question types in KNOWSHIFTQA with their reasoning patterns, example questions, and hypothetical knowledge update illustrated in factual triplets and answers. In example questions, subjects and objects are marked in **bold**, while predicates are marked in *italic*.

gogical approaches (Provenzo et al., 2011), as well as regional and cultural variations in content (Patel, 2015). It remains unclear whether RAG systems can robustly incorporate knowledge from authoritative sources and generate consistent answers under such knowledge discrepancies (scenario illustrated in Figure 1).

To fill this gap, we aim to systematically assess the robustness of RAG systems in performing question answering in K-12 education when encountering knowledge shifts. We present **KNOWSHIFTQA**, a new dataset containing 3,005 multiple-choice questions covering the subjects of Physics, Chemistry, Biology, Geography, and History from the middle-school curriculum. To simulate the factuality discrepancy between LLMs and textbooks, we conduct a *hypothetical knowledge update*, in which we modify the original factual knowledge from the textbooks into plausible alternatives while maintaining coherent and consistent context. Moreover, we tailored a comprehensive question typology to stress-test the *context utilization* and *knowledge integration* abilities of LLMs under such scenarios.

We conducted extensive experiments with various retrieval methods and LLMs. Our findings indicate that most RAG systems exhibit a considerable performance degradation under knowledge shifts. Notably, while many LLMs can effectively incorporate distant contextual facts, they struggle to seamlessly integrate their parametric knowledge with this contextual information. In terms of retrieval, traditional lexical-based methods demonstrate advantages due to the specificity of academic terms; their performance can be further enhanced through an ensemble re-ranking mechanism. Supervised retrieval methods (e.g., Contriever), when fine-tuned directly on our dataset, yield superior performance. Overall, our results highlight the fragility of current RAG systems under knowledge shifts and the challenges LLMs face in effectively combining contextual and parametric knowledge for complex question answering.

2 The KNOWSHIFTQA Dataset

In this section, we introduce the methodology of hypothetical knowledge update and the design of our question typology. The curation pipeline and detailed statistics of the KNOWSHIFTQA dataset are provided in Appendix B and E, respectively.

2.1 Hypothetical Knowledge Update

One of the core objectives of our dataset is to simulate the knowledge discrepancy between LLMs and authoritative textbooks when performing educational QA. However, such discrepancies are often

fuzzy and highly sparse in real-world data, making it infeasible to collect and organize. Consequently, we designed the methodology of hypothetical knowledge update, performing it on high-quality open-source textbooks. The general procedures are as follows: 1) Curate factual questions from textbook paragraphs following our designed question typology. 2) Select a plausible but factually incorrect answer as the updated ground-truth answer. 3) Replace all occurrences of the original answer in the paragraph with the updated answer and adjusted other relevant statements in the context to ensure that the updated paragraph is coherent and consistent. This process is further guaranteed through extensive human curation and verification. Examples of hypothetical knowledge updates in our dataset are provided in Appendix D.

2.2 Question Typology

We identify two potential challenges for LLMs when performing QA under knowledge shifts: 1) *Context Utilization*: Whether LLMs can identify and utilize the corresponding facts from the context; and 2) *Knowledge Integration*: Whether LLMs can incorporate their own parametric knowledge with contextual facts in question answering. To this end, we designed our question typology, as illustrated in Table 1, to investigate such abilities. The two basic question types, *Simple Direct* and *Multi-hop Direct*, aim to evaluate LLMs on simple factual recall and multi-hop reasoning. Based on these two types, we developed the *Multi-hop Distant* type to evaluate the context utilization ability for distant facts from the passage, and the *Multi-hop Implicit*[2] type to evaluate the knowledge integration ability that combines their own factual knowledge with retrieved facts. Moreover, the *Distant Implicit* type poses a greater challenge by combining both features. To ensure our evaluation of knowledge integration ability is independent of knowledge coverage, we restrict the facts requiring LLMs' own knowledge to be head-knowledge only (Sun et al., 2024).

3 Experiments and Analyses

Typically, RAG systems first conduct document retrieval based on given queries, then perform question answering with LLMs based on the retrieved information loaded in the context. In this section, we comprehensively evaluate and analyze the performance of retrieval methods and LLMs on the KnowShiftQA dataset. For details of all tested methods and models, please refer to Appendix C.

Retrieval Methods	Category	R@1	R@5
TF-IDF (Spärck Jones, 1972)	Lexical	65.82	88.72
BM25 (Robertson et al., 1994)	Lexical	82.73	95.27
SPLADE (Formal et al., 2021)	Lexical/Dense	78.04	90.12
Contriever (Izacard et al., 2022)	Dense	53.18	81.80
Con.-msmarco (Izacard et al., 2022)	Dense	76.17	93.54
Mistral-embed (Mistral, 2023a)	Dense	78.74	95.31
Ada-002 (OpenAI, 2022)	Dense	79.23	95.44
Query Rewrite (Ma et al., 2023)	Pre-Retrieval	78.87	94.21
Hybrid Rerank (BM25 + Ada-002)	Ensemble	<u>84.43</u>	<u>96.04</u>
Contriever (fine-tuned)	Dense+FT	<u>84.19</u>	<u>98.96</u>
Con.-msmarco (fine-tuned)	Dense+FT	**87.95**	**99.50**

Table 2: Document retrieval performances (in recall @1/5 %) of retrieval methods from different categories in the KnowShiftQA dataset. The highest scores are marked as **bold**, while the 2nd and 3rd-best scores are <u>underlined</u>. The retrieval granularity is set to paragraph.

3.1 Retrieval Performance

Experimental results of the retrieval methods are presented in Table 2. Traditional lexical retrieval methods, such as BM25, demonstrated strong performance on our dataset, while dense retrieval methods, such as Mistral-embed and Ada-002, achieved comparable performance. Since our dataset focuses on the K-12 education domain, lexical retrieval effectively captures domain-specific keywords in the queries and identifies the corresponding documents. This characteristic underscores the need for fine-tuning dense retrieval models on our dataset. To this end, we fine-tuned both Contriever and Contriever-msmarco on our documents, resulting in significant improvements and highlighting the importance of corpus-specific fine-tuning in educational document retrieval.

Moreover, ensemble methods have demonstrated their effectiveness in various information retrieval tasks (Thakur et al., 2021). We implemented a hybrid approach that retrieves the top k[3] documents with Ada-002, followed by re-ranking with BM25. This ensemble method marginally improved the retrieval performance in both metrics. Conversely, query rewriting[4] (Ma et al., 2023) implemented upon BM25 did not yield performance gains. Detailed results across subjects and question types are provided in Appendix F.

[2]The word 'implicit' indicates that the questions indirectly query the updated facts by embedding them within the middle of the multi-hop reasoning chain.

[3]We use $k = 6$ as the optimal hyperparameter setting.

[4]We use `Mistral-small-2409` as the LLM query rewriter.

Large Language Models	Question Typology					Average
	Simple Direct	*Multi-hop Direct*	*Multi-hop Distant*	*Multi-hop Implicit*	*Distant Implicit*	
Mistral-7b (Mistral, 2023b)	77.70	69.31	72.74	45.32	33.98	61.26
Mixtral-8x22b (Mistral, 2023c)	84.10	84.58	87.15	73.86	65.37	79.67
Mistral-small-2409 (Mistral, 2024a)	87.87	88.70	89.69	72.18	60.81	80.77
Mistral-large-2407 (Mistral, 2024b)	83.93	83.51	87.29	82.25	70.57	81.66
Gemini-1.5-flash (Google, 2024a)	80.98	82.44	87.57	76.02	63.25	78.54
Gemini-1.5-pro (Google, 2024b)	86.56	87.63	88.42	76.26	63.58	81.10
Llama3-8b (Meta, 2024)	90.33	88.85	89.69	63.55	49.43	77.77
Llama3-70b (Meta, 2024)	96.72	96.49	96.89	79.14	63.25	87.42
GPT-3.5-turbo (OpenAI, 2022)	92.62	90.53	91.24	71.22	57.72	81.73
GPT-4 (OpenAI, 2023a)	89.51	89.47	90.68	78.66	70.89	84.46
GPT-4-turbo (OpenAI, 2023b)	95.74	96.18	96.19	81.06	71.71	88.99
GPT-4o (OpenAI, 2024a)	91.97	94.81	93.64	81.29	70.41	87.09
Claude-3-sonnet (Anthropic, 2024a)	94.59	92.82	93.64	77.70	60.65	84.69
Claude-3.5-sonnet (Anthropic, 2024b)	**97.54**	96.49	95.62	83.69	73.82	90.08
o1-mini (OpenAI, 2024b)	95.90	95.73	97.03	85.85	75.45	90.55
o1-preview (OpenAI, 2024b)	95.08	**97.71**	**97.46**	**86.33**	**78.86**	**91.68**

Table 3: Question answering performances (in accuracy %) of LLMs in the KNOWSHIFTQA benchmark with corresponding documents provided. The highest scores are marked as **bold**, while the 2nd and 3rd-best scores are underlined. All LLMs are tested under zero-shot settings, with a *Locate-and-Answer* prompting approach that facilitates active information acquisition from contextual documents, with details in Appendix G.

RAG System	Hypothetical Knowledge Update		Drop
	Before	After	
Llama3-8b + Ada-002	87.49	62.60	24.89
Llama3-8b + Rerank	88.49	66.02	22.47
GPT-4o + Ada-002	96.57	69.65	26.92
GPT-4o + Rerank	97.10	73.71	23.39

Table 4: Performance drop of RAG systems with hypothetical knowledge updates in our benchmark.

3.2 Question Answering Performance

Experimental results of LLMs in question answering are presented in Table 3. Most LLMs exhibited comparable performance for question types *Simple Direct*, *Multi-hop Direct*, and *Multi-hop Distant*. This suggests that **both multi-hop reasoning and distant context utilization do not pose significant challenges for modern LLMs**. However, for *Multi-hop Implicit* questions, which requires the integration of contextual and internal knowledge, a substantial performance disparity emerges between smaller open-source LLMs and advanced models. This disparity is further amplified for *Distant Implicit* questions, where the reasoning chains are getting more complexed. The most capable model we tested, o1-preview, attains over 80% accuracy on *Implicit* questions, whereas Mistral-7b's performance falls below 40%. These findings indicate that **context-memory knowledge integration is an emergent capability presenting greater difficulties for LLMs under knowledge discrepancies**, particularly when coupled with the need for distant context utilization.

3.3 Overall Performance

How do knowledge shifts affect the performance of RAG systems in educational question-answering applications? To answer this question, we selected two representative LLMs: Llama3-8b from open-source models and GPT-4o from proprietary models. These were combined with two high-performing retrieval methods: Ada-002 and hybrid rerank. The resulting RAG systems were tested on questions *before* and *after* hypothetical knowledge updates, with the results presented in Table 4. We observed a significant accuracy drop of 22–27% after these knowledge shifts, indicating substantial performance degradation in modern RAG systems when faced with such scenarios.

4 Related Work

Retrieval-Augmented Generation Following the categorization by Gao et al. (2024), the RAG methods employed in our experiment fall into the categories of *Naive RAG* (lexical, dense) and *Advanced RAG* (rerank, rewrite). Recently, *Modular RAG* has emerged to enhance the adaptability and versatility of RAG systems (Shao et al., 2023; Asai et al., 2023). Furthermore, recent advances in vector databases (Han et al., 2023) and neural graph databases (Bai et al., 2025) highlight the potential for adaptive and generalizable RAG search with structured data (Deng et al., 2024).

Educational Question Answering Prior to the emergence of RAG systems utilizing LLMs, various educational QA systems were developed to provide pedagogically appropriate responses to student inquiries (Abdi et al., 2018; Agarwal et al., 2019). While recent literature explores LLM applications in QA and learning assistance roles (Nye et al., 2023; Kuo et al., 2023; Wang et al., 2024), it is also crucial to leverage logical inference abilities, such as abductive reasoning (Bai et al., 2024; Yim et al., 2024; Zheng et al., 2025b) and inductive reasoning (Qiu et al., 2024; Li et al., 2025; Zheng et al., 2025a), for understanding students' learning outcomes in response to their queries.

Knowledge Discrepancy in LLMs Mitigating knowledge discrepancies or conflicts within LLM applications is a fundamental challenge in LLM research (Xu et al., 2024). Researchers have proposed tuning-based (Li et al., 2022) and prompting-based (Zhou et al., 2023) methods to enhance LLMs' robustness to such conflicts. However, the implications of these conflicts in educational applications and RAG systems remain relatively underexplored. Moreover, it is important to consider the trade-off between factuality robustness (Yu et al., 2024; Zong et al., 2024; Zheng et al., 2025c) and flexibility in instruction following when such discrepancies are present.

5 Conclusion

This paper systematically evaluates the robustness of RAG systems in K-12 educational question answering under knowledge discrepancies using a comprehensive dataset KNOWSHIFTQA. Experimental findings reveal substantial performance degradation in RAG systems when faced with knowledge discrepancies, which is primarily attributed to deficiencies in incorporating contextual and parametric knowledge in question answering—an emergent and challenging ability for modern large language models.

Limitations

We discuss three main limitations of our work.

First, KNOWSHIFTQA employs the approach of hypothetical knowledge updates to effectively simulate real-world knowledge discrepancies for two primary reasons: (1) Real-world knowledge conflicts are often sparse, noisy, and difficult to systematically collect or organize into a cohesive dataset suitable for large-scale evaluation. Hypothetical updates provide a scalable and controlled alternative, allowing us to bypass these limitations. (2) By leveraging a systematic annotation and curation pipeline, we can generate questions with diverse and well-defined reasoning patterns that align with our typology, enabling more robust evaluation of complex question-answering tasks. For future research centered on real-world knowledge discrepancies, we recommend an alternative methodology that incorporates temporal attributes (Chen et al., 2021; Zhang and Choi, 2024). This approach focuses on identifying outdated facts and capturing time-sensitive data (e.g., economic trends, annual events, or societal changes) to construct datasets that reflect real-world knowledge updates. While promising, this method is constrained by the limited overlap between time-sensitive data—often numerical or attribute-specific—and the broader contextual needs of educational question-answering tasks, which may reduce the comprehensiveness of the resulting datasets.

Next, regarding document retrieval, some recent hierarchical retrieval paradigms, such as GraphRAG (Edge et al., 2024) and HippoRAG (Gutiérrez et al., 2024), are not included in our experiments due to their implementation complexity. However, we believe that such structured paradigms could effectively enhance retrieval performance in our scenario, as educational documents are well-structured and contain high-quality factual knowledge.

Finally, this paper primarily evaluates the robustness of RAG systems in the proposed scenario, with experiments conducted using various retrieval methods and large language models. Potential improvements could be achieved through the design of tailored reasoning frameworks via prompting, in-context learning or alignment in LLMs, which we leave for future research.

Ethics Statement

In constructing the KNOWSHIFTQA dataset, we collected text from open-access textbooks. Detailed sources and licenses are provided in Appendix A. The human curation and verification in our annotation pipeline were carried out by a group of postgraduate students with extensive experience in NLP research. We ensured that the updated knowledge is free from harmful or toxic content. It is important to note that our dataset

is designed solely to evaluate the robustness of Retrieval-Augmented Generation systems under scenarios with knowledge discrepancies and is not suitable for assessing the factual accuracy of QA systems.

Acknowledgments

The authors of this paper were supported by the ITSP Platform Research Project (ITS/189/23FP) from ITC of Hong Kong, SAR, China, and the AoE (AoE/E-601/24-N), the RIF (R6021-20) and the GRF (16205322) from RGC of Hong Kong SAR, China.

References

Asad Abdi, Norisma Idris, and Zahrah Ahmad. 2018. Qapd: an ontology-based question answering system in the physics domain. *Soft Computing*, 22:213–230.

A. Agarwal, N. Sachdeva, R. K. Yadav, V. Udandarao, V. Mittal, A. Gupta, and A. Mathur. 2019. Eduqa: Educational domain question answering system using conceptual network mapping. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8137–8141.

Anthropic. 2024a. Blog: Introducing the next generation of claude.

Anthropic. 2024b. Introducing claude 3.5 sonnet.

S. Arbesman. 2012. *The Half-life of Facts: Why Everything We Know Has an Expiration Date*. Human body. Current.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection.

Jiaxin Bai, Yicheng Wang, Tianshi Zheng, Yue Guo, Xin Liu, and Yangqiu Song. 2024. Advancing abductive reasoning in knowledge graphs through complex logical hypothesis generation.

Jiaxin Bai, Zihao Wang, Yukun Zhou, Hang Yin, Weizhi Fei, Qi Hu, Zheye Deng, Jiayang Cheng, Tianshi Zheng, Hong Ting Tsang, Yisen Gao, Zhongwei Xie, Yufei Li, Lixin Fan, Binhang Yuan, Wei Wang, Lei Chen, Xiaofang Zhou, and Yangqiu Song. 2025. Top ten challenges towards agentic neural graph databases.

Jiaxin Bai, Tianshi Zheng, and Yangqiu Song. 2023. Sequential query encoding for complex query answering on knowledge graphs.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions.

Zheye Deng, Chunkit Chan, Weiqi Wang, Yuxi Sun, Wei Fan, Tianshi Zheng, Yauwai Yim, and Yangqiu Song. 2024. Text-tuple-table: Towards information integration in text-to-table generation via global tuple extraction.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization.

Andy Extance. 2023. ChatGPT has entered the classroom: how LLMs could transform education. *Nature*, 623:474–477.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade v2: Sparse lexical and expansion model for information retrieval.

Wensheng Gan, Zhenlian Qi, Jiayang Wu, and Jerry Chun-Wei Lin. 2023. Large language models in education: Vision and opportunities.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Google. 2024a. Gemini flash - google deepmind.

Google. 2024b. Gemini pro - google deepmind.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models.

Yikun Han, Chunjiang Liu, and Pengfei Wang. 2023. A comprehensive survey on vector database: Storage and retrieval technique, challenge.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation.

Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

Bor-Chen Kuo, Frederic T. Y. Chang, and Zong-En Bai. 2023. Leveraging llms for adaptive testing and learning in taiwan adaptive learning platform (talp). In *Empowering Education with LLMs – the Next-Gen Interface and Content Generation*, volume 3487, pages 1–14, Tokyo, Japan. CEUR Workshop Proceedings, CEUR-WS.org.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Chunyang Li, Weiqi Wang, Tianshi Zheng, and Yangqiu Song. 2025. Patterns over principles: The fragility of inductive reasoning in llms under noisy observations.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2022. Large language models with controllable working memory. *arXiv preprint arXiv:2211.05110*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

Mistral. 2023a. Embeddings | mistral ai large language models.

Mistral. 2023b. Mistral 7b | frontier ai in your hands.

Mistral. 2023c. Mixtral of experts | frontier ai in your hands.

Mistral. 2024a. Ai in abundance | mistral ai | frontier ai in your hands.

Mistral. 2024b. Large enough | mistral ai | frontier ai in your hands.

Benjamin D. Nye, Dillon Mee, and Mark G. Core. 2023. Generative large language models for dialog-based tutoring: An early consideration of opportunities and concerns. In *Empowering Education with LLMs – the Next-Gen Interface and Content Generation*, volume 3487, pages 1–15, Tokyo, Japan. CEUR Workshop Proceedings, CEUR-WS.org.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

OpenAI. 2022. New and improved embedding model.

OpenAI. 2023a. Gpt-4 technical report.

OpenAI. 2023b. New models and developer products announced at devday.

OpenAI. 2024a. Hello gpt-4o.

OpenAI. 2024b. Introducing openai o1.

Leigh Patel. 2015. *Decolonizing Educational Research: From Ownership to Answerability*, 1 edition. Routledge, New York.

E.F. Provenzo, A.N. Shaver, and M. Bello. 2011. *The Textbook as Discourse: Sociocultural Dimensions of American Schoolbooks*. Taylor & Francis.

Linlu Qiu, Liwei Jiang, Ximing Lu, Melanie Sclar, Valentina Pyatkin, Chandra Bhagavatula, Bailin Wang, Yoon Kim, Yejin Choi, Nouha Dziri, and Xiang Ren. 2024. Phenomenal yet puzzling: Testing inductive reasoning capabilities of language models with hypothesis refinement. In *The Twelfth International Conference on Learning Representations*.

Srikrishna Raamadhurai, Ryan Baker, and Vikraman Poduval. 2019. Curio SmartChat : A system for natural language question answering for self-paced k-12 learning. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 336–342, Florence, Italy. Association for Computational Linguistics.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*. NIST.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy.

Teotino Gomes Soares, Azhari Azhari, Nur Rokhman, and E Wonarko. 2021. Education question answering systems: a survey. In *Proceedings of The International MultiConference of Engineers and Computer Scientists*.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Kai Sun, Yifan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2024. Head-to-tail: How knowledgeable are large language models (LLMs)? A.K.A. will LLMs replace knowledge graphs? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 311–325, Mexico City, Mexico. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, and Qingsong Wen. 2024. Large language models for education: A survey and outlook.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge conflicts for llms: A survey.

Lixiang Yan, Lele Sha, Linxuan Zhao, Yuheng Li, Roberto Martinez-Maldonado, Guanliang Chen, Xinyu Li, Yueqiao Jin, and Dragan Gašević. 2024. Practical and ethical challenges of large language models in education: A systematic scoping review. *British Journal of Educational Technology*, 55(1):90–112.

Yauwai Yim, Chunkit Chan, Tianyu Shi, Zheye Deng, Wei Fan, Tianshi Zheng, and Yangqiu Song. 2024. Evaluating and enhancing llms agent based on theory of mind in guandan: A multi-player cooperative game under imperfect information.

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, Chunyang Li, Zheyuan Zhang, Yushi Bai, Yantao Liu, Amy Xin, Nianyi Lin, Kaifeng Yun, Linlu Gong, Jianhui Chen, Zhili Wu, Yunjia Qi, Weikai Li, Yong Guan, Kaisheng Zeng, Ji Qi, Hailong Jin, Jinxin Liu, Yu Gu, Yuan Yao, Ning Ding, Lei Hou, Zhiyuan Liu, Bin Xu, Jie Tang, and Juanzi Li. 2024. Kola: Carefully benchmarking world knowledge of large language models.

Michael J. Q. Zhang and Eunsol Choi. 2024. Mitigating temporal misalignment by discarding outdated facts.

Tianshi Zheng, Jiaxin Bai, Yicheng Wang, Tianqing Fang, Yue Guo, Yauwai Yim, and Yangqiu Song. 2024. Clr-fact: Evaluating the complex logical reasoning capability of large language models over factual knowledge.

Tianshi Zheng, Yixiang Chen, Chengxi Li, Chunyang Li, Qing Zong, Haochen Shi, Baixuan Xu, Yangqiu Song, Ginny Y. Wong, and Simon See. 2025a. The curse of cot: On the limitations of chain-of-thought in in-context learning.

Tianshi Zheng, Jiayang Cheng, Chunyang Li, Haochen Shi, Zihao Wang, Jiaxin Bai, Yangqiu Song, Ginny Y. Wong, and Simon See. 2025b. Logidynamics: Unraveling the dynamics of logical inference in large language model reasoning.

Tianshi Zheng, Zheye Deng, Hong Ting Tsang, Weiqi Wang, Jiaxin Bai, Zihao Wang, and Yangqiu Song. 2025c. From automation to autonomy: A survey on large language models in scientific discovery.

Tianshi Zheng, Jiazheng Wang, Zihao Wang, Jiaxin Bai, Hang Yin, Zheye Deng, Yangqiu Song, and Jianxin Li. 2025d. Enhancing transformers for generalizable first-order logical entailment.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models.

Qing Zong, Zhaowei Wang, Tianshi Zheng, Xiyu Ren, and Yangqiu Song. 2024. Comparisonqa: Evaluating factuality robustness of llms through knowledge frequency control and uncertainty.

A Source Textbooks

The documents in the KNOWSHIFTQA dataset is organized based on the following public textbooks (the detailed topics included in each subjects are illustrated in Figure 2.):

Physics *Physics* by Openstax (CC-BY-4.0[5]) https://openstax.org/details/books/physics

Chemistry *Chemistry* by Openstax (CC-BY-4.0[1]) https://openstax.org/details/books/chemistry-2e

Biology *Biology* by Openstax (CC-BY-4.0[1]) https://openstax.org/details/books/biology-2e

History *World History* by OER Commons (CC-BY-NC-4.0[6]) https://oercommons.org/courses/world-history-2

Geography *World Regional Geography* by Sailor Academy (CC-BY-3.0[7]) https://learn.saylor.org/course/view.php?id=722

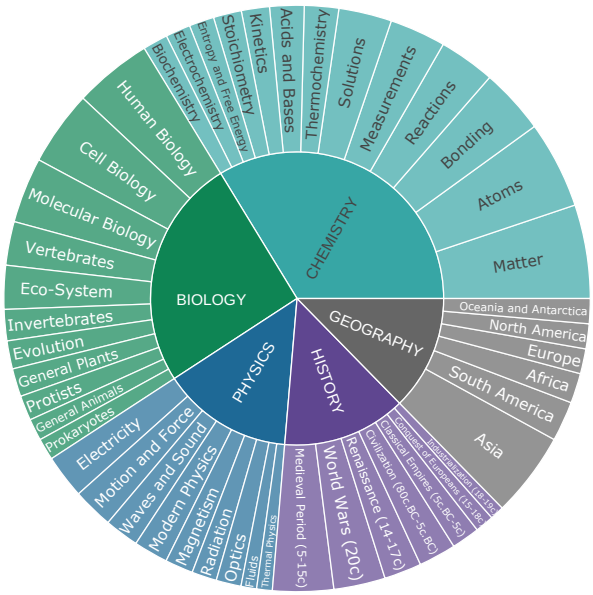


Figure 2: Distribution of five subjects and their corresponding topics included in the KNOWSHIFTQA dataset.

[5]https://creativecommons.org/licenses/by/4.0/deed.en
[6]https://creativecommons.org/licenses/by-nc/4.0/deed.en
[7]https://creativecommons.org/licenses/by/3.0/

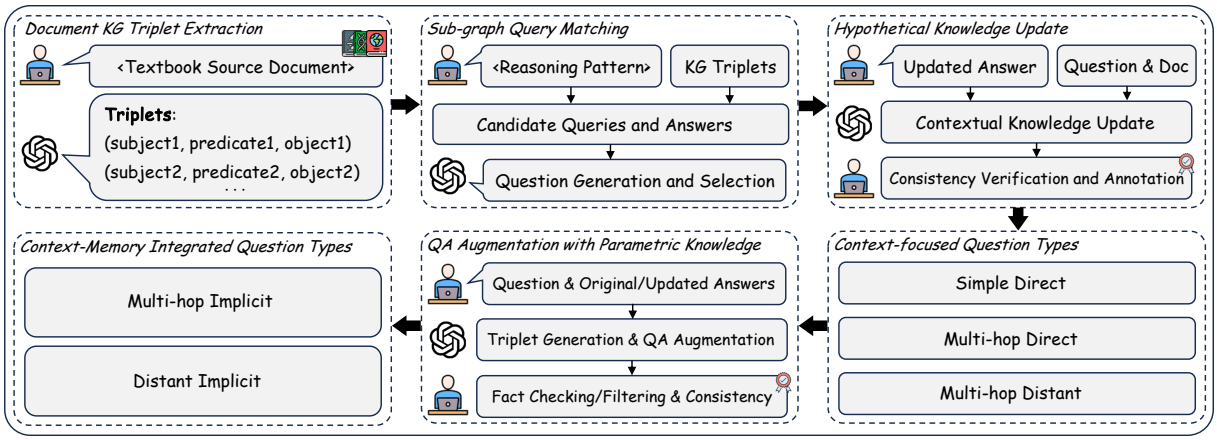


Figure 3: An overview of the data curation pipeline of the KNOWSHIFTQA dataset.

B Curation Pipeline

The curation pipeline of our dataset is illustrated in Figure 3. We first perform triplet extraction on the textbook documents and generate a document-level knowledge graph (KG). Next, we perform sub-graph matching based on fixed reasoning patterns to sample candidate queries, and selectively transform them into natural language questions. Then, hypothetical knowledge update is executed and verified to guarantee consistency between the updated answer and the document.

Context-focused question types, including *Simple Direct*, *Multi-hop Direct*, and *Multi-hop Distant*, are acquired through this process. For *Multi-hop Distant* questions, we leverage distant facts, defined as connected triplet pairs that are separated in the document's sequential ordering. These questions are only assigned to documents containing no fewer than 200 words. To generate the other two *Implicit* question types that require knowledge integration, we perform extra QA augmentation followed by an expert verification process. The design of our question typology is motivated by studies in complex logical reasoning (Bai et al., 2023; Zheng et al., 2024, 2025d).

Our data curation process is performed through an integrated framework involving both human annotators and LLMs. For LLM annotation, we adopted `Claude-3.5-Sonnet` for its outstanding instruction-following ability. Following manual verification, 90.5% of these queries were retained or underwent minor refinements to become high-quality questions, yielding an overall success rate of 86.4%. The total API cost for data annotation is approximately 300 USD.

C Model Details

In this section, we briefly introduce all tested retrieval methods and large language models in our benchmark experiment.

In the retrieval stage of our experiments, we employ a diverse range of retrieval methods. For traditional lexical retrieval, we include **TF-IDF** (Spärck Jones, 1972), which vectorizes documents and queries based on term frequency and inverse document frequency, and **BM25** (Robertson et al., 1994), which enhances TF-IDF with document length normalization and probabilistic term weighting. We also incorporate **SPLADE** (Formal et al., 2021), a method that bridges lexical and dense retrieval paradigms. For dense retrieval, we evaluate several methods that encode questions and documents into the same vector space: **Contriever** (Izacard et al., 2022), an unsupervised text encoder; its fine-tuned variant **Contriever-msmarco**, which we further enhanced by applying contrastive learning (Izacard et al., 2022) to fine-tune both models on the KNOWSHIFTQA dataset for improved retrieval capability; and two closed-source embedding models, **Mistral-embed** (Mistral, 2023a) and **Ada-002** (OpenAI, 2022). Additionally, we explore a pre-retrieval method, **Query Rewrite** (Ma et al., 2023), which reformulates queries to improve retrieval performance. Finally, we implement a **Hybrid Rerank** approach that combines BM25 and Ada-002, leveraging the strengths of both lexical and dense retrieval methods.

In our evaluation, we employ a diverse set of state-of-the-art large language models to assess their performance across various tasks. The models include: Mistral AI's open-source models, ranging from

Original Paragraph	... The halophiles, which means "salt-loving", live in environments with high levels of salt. They have been identified in the Great Salt Lake in Utah and in the Dead Sea between Israel and Jordan, which have salt concentrations several times that of the oceans. ...
Updated Paragraph	... The halophiles, which means "***pressure***-loving", live in environments with high levels of ***pressure***. They have been identified in the ***Mariana Trench*** in ***western Pacific Ocean***, which have higher ***pressure*** than other environments. ...
Question	In which type of environments do halophiles typically live? A. High Acidity Environments. B. High Salt Environments. C. High Pressure Environments. D. High Sugar Environments.
Original Answer	B. High Salt Environments.
Updated Answer	***C. High Pressure Environments.***

Table 5: An example of hypothetical knowledge update for a question in Biology. The modifications of factual knowledge and contextual information in the paragraph are highlighted in ***red***.

the compact **Mistral-7b** (Mistral, 2023b) to the more advanced MoE model **Mixtral-8x22b** (Mistral, 2023c), and their latest iterations **Mistral-small-2409** (Mistral, 2024a) and **Mistral-large-2407** (Mistral, 2024b). Google's Gemini models are represented by **Gemini-1.5-flash** (Google, 2024a) and **Gemini-1.5-pro** (Google, 2024b). Meta's **Llama3** series is included with both 8b and 70b parameter versions (Meta, 2024). We also evaluate OpenAI's models, including **GPT-3.5-turbo** (OpenAI, 2022), **GPT-4** (OpenAI, 2023a), **GPT-4-turbo** (OpenAI, 2023b), and **GPT-4o** (OpenAI, 2024a). Anthropic's LLMs are represented by **Claude-3-sonnet** (Anthropic, 2024a) and **Claude-3.5-sonnet** (Anthropic, 2024b). Lastly, we include OpenAI's latest o1 series, **o1-mini** and **o1-preview** (OpenAI, 2024b), which achieve remarkable performance across various metrics through inference-time scaling. This comprehensive selection allows us to compare a wide range of model architectures and sizes, providing valuable insights into the current state of LLM capabilities.

D Dataset Example

We provide an example question and its corresponding paragraph before and after hypothetical knowledge update in Table 5.

E Dataset Statistics

The statistics for the documents and questions in the KNOWSHIFTQA dataset are provided in Table 6 and Table 7, respectively.

Document Statistics	Subjects					Total
	Chem.	*Bio.*	*Phys.*	*Geo.*	*Hist.*	
Average Document Length	315.5	588.1	647.3	429.8	277.1	437.3
Num of Documents	291	671	166	471	606	2205

Table 6: Average length (in word counts) and quantities of documents in different subjects.

# Questions (avg. length)	Question Types					Total
	Simple Direct	*Multi-hop Direct*	*Multi-hop Distant*	*Multi-hop Implicit*	*Distant Implicit*	
Chemistry	73 (12.4)	88 (17.8)	78 (17.9)	40 (17.4)	68 (21.1)	347 (17.3)
Biology	148 (11.1)	170 (16.7)	248 (16.7)	94 (16.7)	213 (20.7)	873 (16.7)
Physics	46 (12.0)	49 (16.7)	49 (15.6)	41 (16.9)	39 (19.9)	224 (16.1)
Geography	141 (12.1)	147 (16.7)	162 (16.8)	96 (17.7)	144 (21.3)	690 (16.9)
History	202 (12.8)	201 (16.1)	171 (17.0)	146 (18.1)	151 (21.6)	871 (16.8)
Total	610 (12.1)	655 (16.7)	708 (16.9)	417 (17.5)	615 (21.0)	3005 (16.8)

Table 7: Average question length (in word counts) and quantities of questions in different subjects and types.

F Result Details

The comprehensive evaluation results for retrieval approaches and large language models across various subjects and question types are presented in Table 8 and Table 9, respectively. We observed that the performance of lexical retrieval methods correlates positively with question length, while dense retrieval methods exhibit an inverse relationship. This finding suggests that there is potential for developing more sophisticated ensemble methodologies that could fully leverage the strengths of both approaches.

Retrieval Methods	Subjects					Question Types					Average
	chem.	*bio.*	*phys.*	*geo.*	*hist.*	*Sim. Dir.*	*Mul. Dir.*	*Mul. Dis.*	*Mul. Imp.*	*Dis. Imp.*	
BM25	87.90	84.65	74.55	78.26	81.52	79.34	85.04	86.30	75.78	84.23	82.73
Mistral-embed	84.73	77.09	83.48	73.48	80.94	82.13	79.08	79.38	80.81	72.85	78.74
Ada-002	84.44	79.50	82.14	72.75	81.29	82.30	80.76	80.08	80.10	73.01	79.23

Table 8: Retrieval performance (in recall@1 %) of retrieval approaches in different subjects and question types.

Large Language Models	Subjects					Question Types					Average
	chem.	*bio.*	*phys.*	*geo.*	*hist.*	*Sim. Dir.*	*Mul. Dir.*	*Mul. Dis.*	*Mul. Imp.*	*Dis. Imp.*	
Llama3-8b	77.81	75.60	75.45	80.14	78.65	90.33	88.85	89.69	63.55	49.43	77.77
GPT-3.5-turbo	78.93	80.53	82.59	83.04	82.43	92.62	90.53	91.24	71.22	57.72	81.73
GPT-4-turbo	85.30	89.23	86.16	90.00	90.13	95.74	96.18	96.19	81.06	71.71	88.99

Table 9: Performance (in accuracy %) of LLMs in question answering in different subjects and question types.

G Prompting Approach

In our experiments, we adopt the prompting approach of *Locate-and-Answer*, to facilitate active acquisition of information in the context when performing question answering. We first request LLMs to identify and locate the corresponding sentence that include the knowledge for the question from the provided document, and then reason to provide its answer. According to the experimental result in Table 10, this prompting approach can effectively improve the QA performance of LLMs compared to direct answering.

Model	**Prompting Method**	
	Direct Answer	Locate-and-Answer
Gemini-1.5-flash	75.21	78.54
GPT-3.5-turbo	73.91	81.73
GPT-4o	80.43	87.09

Table 10: Performance (in accuracy %) of LLMs in question answering with different prompting methods.

H Calibration, Parametric Knowledge, and Instruction Following in RAG Systems

In our study, retrieved documents serve as the unequivocal reference, mirroring the pedagogical approach in K-12 education where textbooks are authoritative. However, our evaluation results reveal counterintuitive performance patterns in state-of-the-art LLMs, particularly concerning simple direct questions. Stronger models, such as GPT-4 and GPT-4o, occasionally underperform compared to weaker ones on these questions, which primarily assess direct factual recall from the provided (and hypothetically updated) context. This suggests that the calibration of these advanced LLMs might lead them to exhibit excessive confidence in their internal, parametric knowledge, thereby inhibiting their consistent reliance on the external documents explicitly provided as the ground truth for the task.

This observation highlights a critical tension inherent in RAG systems: the trade-off between an LLM's adherence to its ingrained parametric knowledge (what might be termed its internal "factuality calibration") and its capacity for robust instruction following. While an LLM prioritizing its internal knowledge, especially if it aligns with generally accepted facts, could be interpreted as a form of "honesty," this perspective shifts within the specific operational context of RAG systems like those evaluated with KNOWSHIFTQA. In such systems, LLMs are explicitly instructed to base their responses on the provided contextual documents. Therefore, when a model disregards this instruction and favors its parametric knowledge—particularly when the context contains deliberate, hypothetical updates—it signifies a limitation in its instruction-following capability rather than a commendable adherence to broader factual accuracy.

The underperformance on simple direct questions can thus be partly attributed to this conflict: the models may struggle to override a strong internal "prior" when explicitly instructed to use new, conflicting information from the context. This distinction is paramount for developing robust educational QA systems. Such systems must reliably reflect the content of designated authoritative sources (e.g., textbooks), even if these sources present information that has been updated or differs from an LLM's training data. Consequently, the ability to flexibly integrate and prioritize instructed contextual information is a fundamental requirement for their effectiveness in dynamic knowledge environments. While in real-world scenarios with less clearly defined ground truths or more ambiguous knowledge conflicts, the sophisticated reasoning of stronger LLMs might offer advantages in navigating discrepancies, the specific demands of an educational RAG system underscore the critical importance of faithful instruction adherence.

Improving Parallel Sentence Mining for Low-Resource and Endangered Languages

Shu Okabe[1,2] **and Katharina Hämmerl**[1,2] **and Alexander Fraser**[1,2,3]
[1]School of Computation, Information and Technology
Technische Universität München (TUM)
[2]Munich Center for Machine Learning
[3]Munich Data Science Institute
{shu.okabe,k.haemmerl,alexander.fraser}@tum.de

Abstract

While parallel sentence mining has been extensively covered for fairly well-resourced languages, pairs involving low-resource languages have received comparatively little attention. To address this gap, we present BELOPSEM, a benchmark of new datasets for parallel sentence mining on three language pairs where the source side is low-resource and endangered: Occitan-Spanish, Upper Sorbian-German, and Chuvash-Russian. These combinations also reflect varying linguistic similarity within each pair. We compare three language models in an established parallel sentence mining pipeline and apply two types of improvements to one of them, Glot500. We observe better mining quality overall by both applying alignment post-processing with an unsupervised aligner and using a cluster-based isotropy enhancement technique. These findings are crucial for optimising parallel data extraction for low-resource languages in a realistic way.

1 Introduction

Parallel sentence mining aims to find matching sentence pairs from a source and a target language. This task is usually the first step in creating a corpus to train other NLP models, most famously Machine Translation (MT) systems. So far, significant progress has been reached for well-resourced languages and language pairs, such as in the BUCC Shared Task 2017 (Zweigenbaum et al., 2017), where English was paired with four other high-resource languages.[1] However, when one of the languages is low-resource and hence not yet well supported by the language models, similar performance cannot be achieved. Besides, sentence encoding models heavily rely on parallel data during pre-training. Without enough parallel sentences, decent sentence representation cannot be reached. For instance, Tan et al. (2023) rely on at least tens of thousands of *parallel* sentences in their sentence representation. Therefore, before reaching these parallel data size milestones, sentence encoders are not reliable enough for mining, which mainly affect very low-resource languages and, more critically, endangered languages.

The alternative is then to rely on language models pre-trained with only monolingual data by averaging word representations. Mining will be more difficult because sentence encoders perform better than mean-pooled embeddings for sentence mining. Thus, we focus on trying to improve such alternative sentence embeddings effectively through two methods: Alignment post-processing and isotropy enhancement in the embedding space. Hangya and Fraser (2019) used alignments with static embeddings to filter sentence pairs, but we updated it with an unsupervised multilingual aligner. The second method has already been shown to be effective on several language pairs on the related but different task of sentence matching (Hämmerl et al., 2023).

Our main contributions are: 1) We create BELOPSEM (**Be**nchmark of **lo**w-resource languages for **p**arallel **se**ntence **m**ining), which consist of three new realistic benchmarks for parallel sentence mining for three language pairs where the source language is low-resource and endangered. 2) We improve on baseline approaches to parallel sentence mining by applying isotropy enhancement and word-level alignment, neither of them requiring additional resources such as parallel sentences, unlike state-of-the-art sentence encoder models such as LaBSE (Feng et al., 2022). We publicly release the benchmark datasets[2] and the updated mining pipeline.[3]

[1]LaBSE (Feng et al., 2022) reaches more than 88 points of F-score on all four BUCC corpora.

[2]At: https://github.com/shuokabe/Belopsem/.

[3]At: https://github.com/shuokabe/PaSeMiLL/.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 196–205
July 27 - August 1, 2025

2 Languages and corpora

2.1 Three language pairs

We focus on the following three language pairs, where the source language is an endangered language according to Ethnologue (Eberhard et al., 2024), and the target language is well-resourced: Occitan-Spanish, Upper Sorbian-German, and Chuvash-Russian. We choose these pairs to represent different mining difficulties with a range in language distance in the pair. They also happen to have received recent attention from the NLP community through WMT Shared Tasks (Libovický and Fraser, 2021; Sánchez-Martínez et al., 2024). We note that all three source languages are classified as 'Scraping-Bys' (1) in the classification of Joshi et al. (2020), i.e. low-resource.

Occitan-Spanish (OCI-ES) Occitan (ISO code: `oci`; Glottocode: `occi1239`) is spoken in the south of France, Italy, and Spain. Both Occitan and Spanish belong to the Western Romance language family and are written in the Latin script. This pair represents a close language pair which can heavily rely on related languages (French or Spanish directly).

Upper Sorbian-German (HSB-DE) Upper Sorbian (`hsb`; `uppe1395`) is a West Slavic language spoken in Saxony in Germany, while German belongs to the Germanic family. Hence, both are Indo-European languages and use the same Latin script. This pair is more challenging since the distance between the two languages is larger. Yet, Upper Sorbian can rely on related Slavic languages, namely Czech and Polish, which are better resourced and supported by existing language models.

Chuvash-Russian (CHV-RU) Chuvash (`chv`; `chuv1255`) is a Turkic language spoken in the Russian Federation, while Russian belongs to the Slavic language family. Both are written in the Cyrillic alphabet. The language distance for the pair is thus larger than the previous two pairs, which makes it the most challenging of the three.

2.2 Parallel sentence mining corpus creation

We follow the overall methodology used for the BUCC 2017 Shared Task (Zweigenbaum et al., 2017) to create a controlled experimental environment: we inject sentences from existing parallel sentences into monolingual corpora. We mainly use the Leipzig corpora (Goldhahn et al., 2012) for monolingual sentences and choose similar sources (e.g., same year of Wikipedia or news) for both languages in a pair. Parallel sentences for Upper Sorbian-German and Chuvash-Russian come from two editions of the WMT Shared Task in Unsupervised MT and Very Low Resource Supervised MT (Fraser, 2020; Libovický and Fraser, 2021), while we use parallel data from Wikimedia 2023, available on OPUS (Tiedemann, 2012), for Occitan-Spanish. We pre-process the sentences to ensure encoding consistency and filter out too-short sentences. We aim for 6% of true parallel sentences in the final corpora. Appendix A presents more details on the data curation.

Table 1 presents the number of sentences in the corpora of each language pair. We split the generated corpora to have 25% assigned to the training dataset and the remaining for testing. In the original BUCC Shared Task, the training and testing balance was 50:50. However, we chose a 25:75 split to simulate a scenario where we want to mine further parallel sentences from larger corpora.

	train	test
Occitan corpus	7,899	23,675
Spanish corpus	7,780	23,334
of which parallel	486	1,457
Upper Sorbian corpus	15,980	47,847
German corpus	15,999	47,999
of which parallel	1,000	3,000
Chuvash corpus	7,998	23,995
Russian corpus	7,994	23,985
of which parallel	499	1,498

Table 1: Number of sentences in the datasets for all three language pairs in BELOPSEM.

3 Parallel sentence mining methodology

3.1 Standard mining pipeline

We follow PASEMILL (Okabe and Fraser, 2025), which is similar to the established mining pipeline of Hangya and Fraser (2019), but updates it using contextual language models instead of static embeddings. First, we encode both source and target sentences in the same sentence representation space using a multilingual language model. When relying on mean-pooled sentence representations, we choose the 8th layer in line with previous findings (Hämmerl et al., 2023; Imani et al., 2023; Okabe and Fraser, 2025).

Then, we use the CSLS (Cross-Domain Similarity Local Scaling) score (Conneau et al., 2018) to compute the similarity between a source and target sentence. The CSLS score is related to established margin-based methods (Artetxe and Schwenk, 2019) and is defined as follows:

$$\text{CSLS}(x, y) = 2\cos(x, y) - \sum_{z \in \text{NN}_k(x)} \frac{\cos(x, z)}{k} - \sum_{z \in \text{NN}_k(y)} \frac{\cos(y, z)}{k}, \quad (1)$$

where $\text{NN}_k(x)$ indicates the k-nearest neighbours of vector x. In our experiments, we set $k = 20$. CSLS is known to suffer less from the hubness problem (Dinu et al., 2015) and to get better performance compared to the standard cosine similarity (Artetxe and Schwenk, 2019). In our preliminary work on the Upper Sorbian-German pair, we did not find major differences in mining, whether we used the margin-based score or the CSLS score to find the nearest neighbour.

Finally, we pair each source sentence with its closest target sentence. The actual computation of the nearest neighbours is carried out using the Faiss library (Johnson et al., 2019). Since not all source sentences have a matching counterpart, we filter the mined pairs according to a threshold θ. We define this similarity threshold dynamically, as in Hangya et al. (2018):

$$\theta = \text{mean}(S) + \lambda \times \sigma(S), \quad (2)$$

where we compute the mean and standard deviation (σ) values of the similarity scores (S), with λ as a hyperparameter. Details on computational efficiency can be found in Appendix B.

3.2 Multilingual language models

To represent the sentences, we stick to three language models in our analysis. First, we select a multilingual language model which is widely used: XLM-RoBERTa or XLM-R (base) (Conneau et al., 2020). The second model is LaBSE (Feng et al., 2022), a state-of-the-art sentence embedding model. Our preliminary experiments showed better performance of LaBSE over LASER (Artetxe and Schwenk, 2019; Costa-jussà et al., 2022) on our datasets. We do not fine-tune sentence encoders due to their need for *parallel* sentences because it is not accessible for languages which do not yet have such a sizeable corpus. None of the two models have seen sentences in Occitan, Upper Sorbian, or Chuvash; they can only rely on related languages for the mining process (e.g., other Romance languages for Occitan, Polish or Czech for Upper Sorbian, or Kazakh for Chuvash).

The main language model we work on is Glot500-m (Imani et al., 2023), which further pretrains XLM-R (base) with *monolingual* data for more than 500 low-resource languages with an extended vocabulary. We choose this model for two reasons: it has seen all three source languages, which means a better representation, and it can be directly compared to XLM-R. An additional point to consider for our three source languages is the data support from Glot500. Occitan is already well supported by the model (among the top 50 tail languages,[4] with more than 1M sentences). Fewer (100k) sentences have been seen for Upper Sorbian, which lies in the mid-to-low pre-training rank of Glot500. Finally, while representing Chuvash solely with related languages may be difficult (for XLM-R or LaBSE), better representation can be obtained from Glot500 (800k pre-training sentences).

3.3 Unsupervised alignment post-processing

Comparing the word-level alignment is a method to improve parallel sentence mining, as already documented in Hangya and Fraser (2019), for instance, but there they relied on *static* word embedding similarity. Instead, we use an unsupervised multilingual aligner, SimAlign (Jalili Sabet et al., 2020), which leverages pre-trained language models to compute alignment links. In our case, we use the 8th layer of Glot500 and count the number of alignment links. We choose the `argmax` method for higher-precision links. We divide the number of symmetrised links by the number of words in the source and target sentences to get the average alignment ratio. A full alignment between both sentences hence gets a score of 1.

We additionally filter alignment links according to their overall frequency, keeping only the most frequent (and hence, more reliable) word pairs. Finally, we also select the mined sentence pairs using a dynamic alignment threshold α (as in Equation 2), which depends on the mean and standard deviation of all scores and a hyperparameter μ.

Our preliminary experiments showed that using a lower filtering threshold θ and then applying alignment post-processing yielded a better F-score in the end. Such filtering increases precision; start-

[4] Languages *not* seen by XLM-R during pre-training.

ing from 'noisier' sentence pairs actually led to a quicker gain in precision for reduced loss in recall, and hence an overall improvement of the F-score. Due to this choice, alignment post-processing may lead to higher recall and lower precision than the standard approach (cf. complete results below).

3.4 Cluster-based isotropy enhancement

To enhance isotropy in sentence representations, Hämmerl et al. (2023) explored ZCA whitening (Huang et al., 2021) and cluster-based isotropy enhancement (CBIE; Rajaee and Pilehvar, 2021). Both yielded comparable results on cross-lingual sentence retrieval for 1K sentences. We choose CBIE for its better scalability when it comes to larger datasets, as in our case, and apply it to the mean-pooled sentence representation before a similarity search for each dataset separately. This method determines dominant directions not globally but for each cluster in the space using Principal Component Analysis and removes the top 12 components. Appendix C shows t-SNE visualisations (van der Maaten and Hinton, 2008) of anisotropy for all three language pairs.

We do not report the results of the CBIE transformation on LaBSE because sentence embeddings suffer less from anisotropic representation and benefit little from it, as noted in Hämmerl et al. (2023).

4 Experimental results

We tune the similarity and alignment score thresholds (θ and α), or more exactly, their associated hyperparameters (λ and μ), on each training dataset. We evaluate the experiments using the standard Precision (P), Recall (R), and F-score (F). Table 2 presents the mining results for all three language pairs on the test sets of BELOPSEM. The F-score in **bold** is the overall best score, while the underlined score is the best among variants of Glot500. Complete results with precision and recall scores are available in Appendix D.

From the F-scores of the two baseline models, we see that having closer source and target languages is beneficial, which is in line with our intuition on language distance. The number of sentences in related languages may be another factor: While Occitan can benefit from very well-resourced languages (namely, French or Spanish itself), and Upper Sorbian has close links with Czech and Polish, Chuvash is far from other Turkic languages (even those only supported by Glot500).

LM	align.	CBIE	OCI-ES	HSB-DE	CHV-RU
XLM-R	NO	NO	48.08	1.43	3.06
LaBSE	NO	NO	**93.50**	**67.21**	28.24
Glot500	NO	NO	72.61	20.85	37.84
Glot500	YES	NO	77.19	32.70	37.39
Glot500	NO	YES	83.29	43.15	41.51
Glot500	YES	YES	<u>84.46</u>	<u>50.82</u>	<u>**43.62**</u>

Table 2: F-scores (%) on the test datasets of the three mining corpora in BELOPSEM.

XLM-R's performance also shows that relying only on related languages is not enough to carry out mining with mean-pooled sentence embeddings.

Regarding the standard Glot500 model, we notice better performance than XLM-R due to the additional pre-training (around or more than 20 points in F-score), but the mining quality is still worse than LaBSE. The notable exception is Chuvash, where the additional sentences seem to strongly help the model, gaining more than 9 points.

Using alignment post-processing only brings slight improvement on the final F-score, except for Upper Sorbian. For that corpus, this suggests that a significant number of actual parallel sentence pairs have a rather lower similarity score (and are thus discarded when the similarity threshold θ is too high). This seems to be mainly due to the language representation (i.e., number of pre-training sentences of Glot500), as it does not affect the other two languages as massively.

On the other hand, the CBIE transformation brings consistent improvement to Glot500: Related language pairs seem to benefit more from the transformation (+10 for OCI-ES and +22 for HSB-DE) than distant languages (+4 for CHV-RU). This confirms that the method also enables better representation of low-resource languages in the sentence embedding space. Finally, combining both improvement techniques leads to the best results. Still, the main increase seems to come from the CBIE transformation; the alignment post-processing rather complements it with more detailed filtering.

Qualitative analysis Table 3 presents an example of a sentence pair that was correctly mined after applying the CBIE transformation. We see that our base model pairs the Occitan sentence with a Spanish sentence unrelated to university work, resulting in a negative similarity score; hence, such a pair is discarded. On the other hand, isotropy

	sentence	sim.
OCI	A l'òra d'ara, qu'ensenha a l'Universitat de Nagoya.	
ES	A la fecha sólo sobrevive Pablo, quien vive en Parácuaro, Michoacán. *To date, only Pablo, who lives in Parácuaro, Michoacán, survives.*	-0.004
+ CBIE	Actualmente, trabaja en la Universidad de Nagoya. *He is currently working at Nagoya University.*	0.118

Table 3: Example of sentence mined for the Occitan-Spanish (OCI-ES) corpus before and after CBIE transformation with corresponding similarity scores.

enhancement enables a positive and higher similarity score with the correct Spanish translation of the sentence. The noticeable difference in similarity scores explains why this sentence pair has been mined by the second system. For this very sentence pair, we hypothesise that the named entity 'Universitat de Nagoya' (or Nagoya University) created confusion, coupled with the not obvious association of the four-word phrase 'A l'òra d'ara' with 'Actualmente' (currently).

5 Related works

Parallel sentence mining has mainly been studied to prepare a corpus to train MT models. Yet, few works focus on low-resource language pairs. Kvapilíková and Bojar (2023) already considered mining parallel sentences for the Upper Sorbian-German language pair using their pipeline (Kvapilíková et al., 2020). Their main goal was to crawl pseudo-parallel sentences for MT, for which they observed that 500k Upper Sorbian sentences were necessary to reach acceptable mining quality. Similarly, Heffernan et al. (2022) improve LASER with multilingual distillation (LASER3), requiring large amounts of monolingual data and a significant number of parallel sentences. Tan et al. (2023) rely on contrastive learning to further boost LASER3, but it still needs more than 40,000 *parallel* sentences coupled with back-translation even in the extremely low-resource scenario.

Two shared tasks provided comparable corpora: the BUCC Shared Tasks (Zweigenbaum et al., 2017, 2018), whose methodology we used to generate our corpora, and the Shared Task on corpus filtering for low-resource language pairs (Koehn et al., 2019, 2020). For the latter, the size of parallel corpora reaches several hundreds of thousands, which is difficult to obtain for endangered low-resource languages.

6 Conclusion

Low-resource languages, especially endangered ones, can benefit from bilingual sentence mining to gather a parallel corpus for downstream tasks, such as MT. However, current mining tools can show variable performance, with poorer results when one of the languages is neither supported nor close enough to other languages seen by state-of-the-art sentence encoders, such as LaBSE.

We thus focused on using a model which was trained using only monolingual data, Glot500. For controlled experimental conditions, we generated sentence mining corpora for three language pairs with varying degrees of linguistic similarity (BELOPSEM). We found that the CBIE transformation to enhance isotropy in the sentence representation consistently improves mining quality, while alignment post-processing helps when the language is not well-represented in the language model. These results hint at the possibility for low-resource languages to reach higher mining quality *without parallel data*.

The mining quality improvement we observed can help the early stages of parallel corpus creation where monolingual data is sufficiently present, but there are not enough parallel sentences to fine-tune sentence encoder models. We hope that the benchmark can also foster work on sentence mining for other language pairs. Future work will mainly extend this methodology to other language pairs (different scripts and lower or no support in Glot500).

Limitations

The choice of the language pairs can be discussed as they may not be the most challenging cases, with two of them from the same Indo-European language family and all three pairs using each the same script. Nonetheless, we stuck to plausible pairs with potential needs, having deployment in real-life settings in mind. For research purposes, considering pairs such as Chuvash-English, replacing the Russian target sentences with their translations might be an option to see further linguistic differences.

Moreover, compared to the state-of-the-art LaBSE model, mining performance can be notably lower with our mining pipeline using Glot500 (or any mean-pooled sentence embeddings). We

have, however, seen that for a language that is too far from the languages used to pre-train LaBSE, Glot500 is actually a better alternative for mining (cf. Chuvash-Russian). This means that by pre-training XLM-R with *monolingual* data on the language, we can reach higher mining quality. Eventually, with enough parallel sentences (possibly mined), we can fine-tune LaBSE or similar sentence encoders. The main goal of our work is to lower the data barrier for low-resource language pairs by removing the need for parallel data.

Finally, the size of the datasets remains small for all three settings. We restricted the number of sentences overall to stay in realistic dataset sizes (several tens of thousands of monolingual sentences) while maintaining a parallel sentence ratio of around 6%.

Acknowledgments

We thank Viktor Hangya for his help and the anonymous reviewers for their insightful comments. This work has received funding from the European Research Council (ERC) under grant agreement No. 101113091 - Data4ML, an ERC Proof of Concept Grant.

References

Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2015. Improving zero-shot learning by mitigating the hubness problem. In *Proceedings of the Workshop Track at ICLR*.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig. 2024. *Ethnologue: Languages of the World*, twenty-seventh edition. SIL International, Dallas, Texas. Online version.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Alexander Fraser. 2020. Findings of the WMT 2020 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 765–771, Online. Association for Computational Linguistics.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Katharina Hämmerl, Alina Fastowski, Jindřich Libovický, and Alexander Fraser. 2023. Exploring anisotropy and outliers in multilingual language models for cross-lingual semantic sentence similarity. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7023–7037, Toronto, Canada. Association for Computational Linguistics.

Viktor Hangya, Fabienne Braune, Yuliya Kalasouskaya, and Alexander Fraser. 2018. Unsupervised parallel sentence extraction from comparable corpora. In *Proceedings of the 15th International Conference on Spoken Language Translation*, pages 7–13, Brussels. International Conference on Spoken Language Translation.

Viktor Hangya and Alexander Fraser. 2019. Unsupervised parallel sentence extraction with parallel segment detection helps machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1224–1234, Florence, Italy. Association for Computational Linguistics.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2101–2112, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. WhiteningBERT: An easy unsupervised sentence embedding approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 238–244, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the WMT 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742, Online. Association for Computational Linguistics.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Ivana Kvapilíková, Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Ondřej Bojar. 2020. Unsupervised multilingual sentence embeddings for parallel corpus mining. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 255–262, Online. Association for Computational Linguistics.

Ivana Kvapilíková and Ondřej Bojar. 2023. Boosting unsupervised machine translation with pseudo-parallel data. In *Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track*, pages 135–147, Macau SAR, China. Asia-Pacific Association for Machine Translation.

Jindřich Libovický and Alexander Fraser. 2021. Findings of the WMT 2021 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 726–732, Online. Association for Computational Linguistics.

Shu Okabe and Alexander Fraser. 2025. Bilingual Sentence Mining for Low-Resource Languages: a Case Study on Upper and Lower Sorbian. In *Proceedings of the Eighth Workshop on the Use of Computational Methods in the Study of Endangered Languages*, Honolulu. Association for Computational Linguistics.

Sara Rajaee and Mohammad Taher Pilehvar. 2021. A cluster-based approach for improving isotropy in contextual embedding space. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 575–584, Online. Association for Computational Linguistics.

Felipe Sánchez-Martínez, Juan Antonio Perez-Ortiz, Aaron Galiano Jimenez, and Antoni Oliver. 2024. Findings of the WMT 2024 shared task translation into low-resource languages of Spain: Blending rule-based and neural systems. In *Proceedings of the Ninth Conference on Machine Translation*, pages 684–698, Miami, Florida, USA. Association for Computational Linguistics.

Weiting Tan, Kevin Heffernan, Holger Schwenk, and Philipp Koehn. 2023. Multilingual representation distillation with contrastive learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1477–1490, Dubrovnik, Croatia. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.

Pierre Zweigenbaum, Serge Sharoff, and Reinhard Rapp. 2017. Overview of the second BUCC shared task: Spotting parallel sentences in comparable corpora. In

Proceedings of the 10th Workshop on Building and Using Comparable Corpora, pages 60–67, Vancouver, Canada. Association for Computational Linguistics.

Pierre Zweigenbaum, Serge Sharoff, and Reinhard Rapp. 2018. Overview of the third BUCC shared task: Spotting parallel sentences in comparable corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

A Data curation

We report the original data used to generate our three parallel sentence mining corpora.

Occitan-Spanish For the monolingual sentences, we use the 2021 Wikipedia data from the Leipzig corpora (30K) for both languages (Goldhahn et al., 2012). For the parallel sentences, we use the 2023 Wikimedia data from OPUS[5] (Tiedemann, 2012).

Upper Sorbian-German We use the WMT 2020 Shared Task data for both parallel sentences and monolingual Upper Sorbian data (Fraser, 2020). The monolingual data was provided by the Sorbian Institute. We use both development and test data (`devtest.tar.gz`) for the parallel sentences, which were controlled for the Shared Task evaluation. For German monolingual data, we use the Leipzig news corpora of 2020 (300K sentences).

Chuvash-Russian For the monolingual sentences, we also use the 2021 Wikipedia data from the Leipzig corpora (30K) for both languages. We chose this year to have a better correspondence with the parallel data provided by the WMT 2021 Shared Task (Libovický and Fraser, 2021). From the parallel data, we use the development set (`devel.chv-ru.{chv|ru}`), which was manually filtered for the Shared Task.

Licence Corpora downloaded from the Leipzig corpora are licenced under the Creative Commons Licence CC BY.[6] The Wikimedia data from OPUS has a CC BY-SA licence. For the resources from the WMT Shared Tasks, the Chuvash data has a CC0 licence, while the Upper Sorbian datasets have a CC BY-NC-SA licence. We hence release our corpora with the following licences: the Occitan-Spanish corpus with a CC BY-SA licence, the Upper Sorbian-German corpus with a CC BY-NC-SA licence, and the Chuvash-Russian corpus with a CC BY licence.

[5] https://opus.nlpl.eu/wikimedia/es&oc/v20230407/wikimedia.

[6] https://wortschatz.uni-leipzig.de/en/usage.

B Reproducibility

We ran our experiments on 1 GPU: we mainly used one NVIDIA H100 for faster computation, but also one NVIDIA Tesla V100. It is worth mentioning that Faiss (Johnson et al., 2019) or SimAlign (Jalili Sabet et al., 2020) can also be run on CPUs, albeit slowly, and the CBIE transformation does not require GPUs. In terms of actual computation time, the sentence representation step takes a few minutes on one V100 GPU, and so does the mining process (less than 10 minutes). The alignment post-processing on GPU is also very quick: Since we apply it on mined sentence pairs with a given threshold θ, we only consider a subset of the original monolingual corpora. Finally, the CBIE transformation takes only a few minutes on a CPU.

C Visualisation of anisotropy

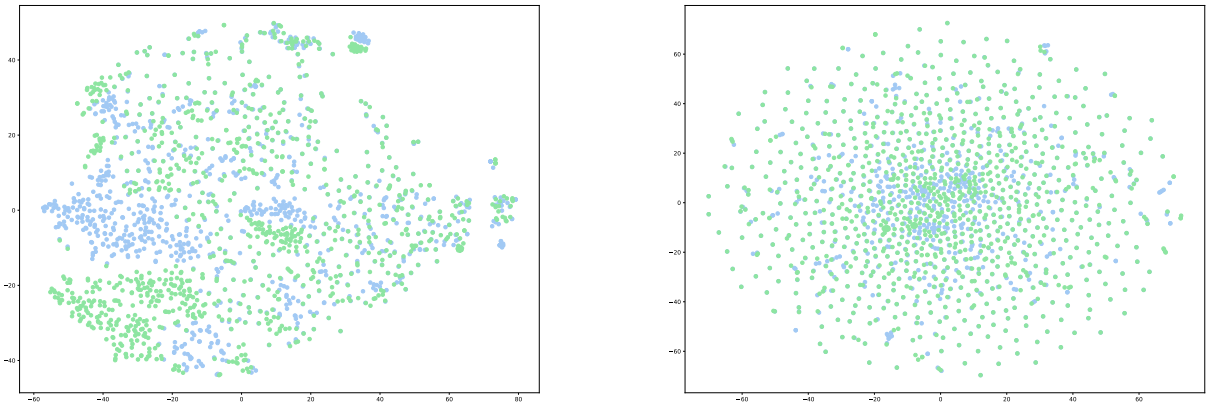
To visualise the impact of the CBIE transformation on the sentence representation space, we also follow the methodology of Hämmerl et al. (2023). We present t-SNE plots of 1,000 parallel sentences from each language pair before and after applying the CBIE transformation in Figures 1, 2, and 3.

As observed in Hämmerl et al. (2023), languages that have been less seen during pre-training suffer more from the issue of anisotropy.

D Complete experimental results

Tables 4, 5, and 6 present the complete mining results for all three language pairs. As in Section 4, **bold** F-score stands for the overall best score, whereas the <u>underlined</u> score is the best among variants of Glot500.

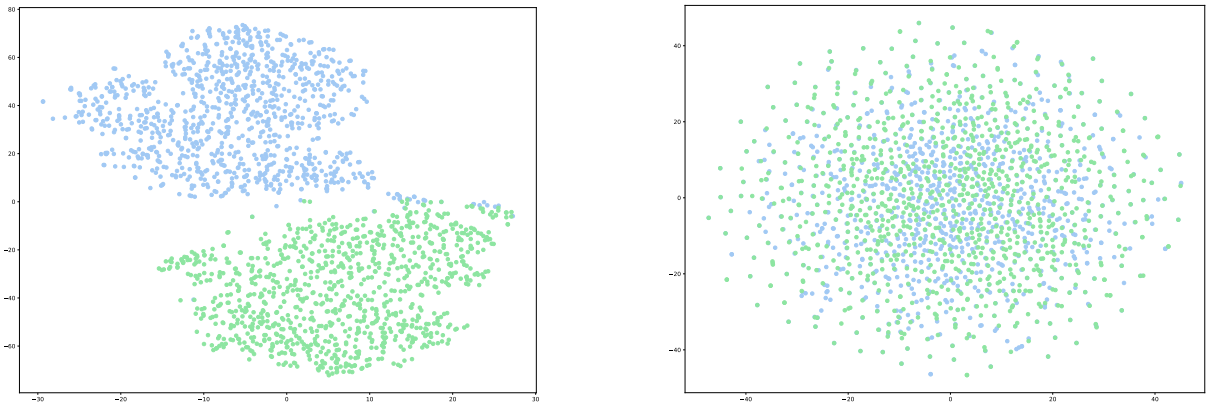
We note here that the CBIE transformation mainly improves precision rather than recall. This implies that enhanced isotropy helps to better filter sentences.



(a) Before transformation

(b) After CBIE transformation

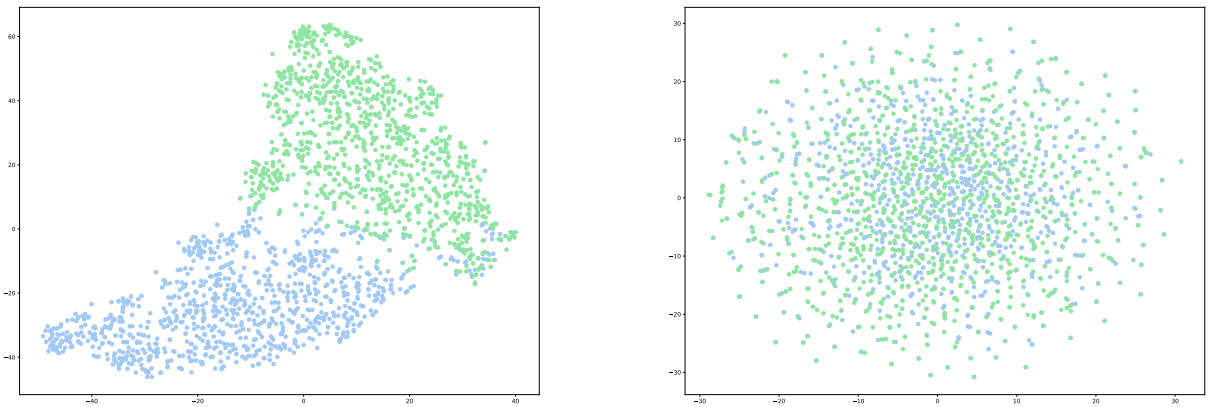
Figure 1: t-SNE plots for 1,000 parallel **Occitan-Spanish** sentences before and after CBIE transformation.



(a) Before transformation

(b) After CBIE transformation

Figure 2: t-SNE plots for 1,000 parallel **Upper Sorbian-German** sentences before and after CBIE transformation.



(a) Before transformation

(b) After CBIE transformation

Figure 3: t-SNE plots for 1,000 parallel **Chuvash-Russian** sentences before and after CBIE transformation.

LM	align.	CBIE	P (%)	R (%)	F (%)
XLM-R	NO	NO	59.18	40.49	48.08
LaBSE	NO	NO	94.72	92.31	**93.50**
Glot500	NO	NO	72.81	72.41	72.61
Glot500	YES	NO	76.36	78.04	77.19
Glot500	NO	YES	90.50	77.14	83.29
Glot500	YES	YES	90.38	79.27	84.46

Table 4: Evaluation on the test dataset of the **Occitan-Spanish** corpus.

LM	align.	CBIE	P (%)	R (%)	F (%)
XLM-R	NO	NO	1.64	1.27	1.43
LaBSE	NO	NO	83.22	56.37	**67.21**
Glot500	NO	NO	22.72	19.27	20.85
Glot500	YES	NO	47.03	25.07	32.70
Glot500	NO	YES	60.74	33.47	43.15
Glot500	YES	YES	61.03	43.53	50.82

Table 5: Evaluation on the test dataset of the **Upper Sorbian-German** corpus.

LM	align.	CBIE	P (%)	R (%)	F (%)
XLM-R	NO	NO	3.84	2.54	3.06
LaBSE	NO	NO	42.18	21.23	28.24
Glot500	NO	NO	41.40	34.85	37.84
Glot500	YES	NO	39.57	35.45	37.39
Glot500	NO	YES	46.67	37.38	41.51
Glot500	YES	YES	54.62	36.32	**43.62**

Table 6: Evaluation on the test dataset of the **Chuvash-Russian** corpus.

Revisiting Epistemic Markers in Confidence Estimation: Can Markers Accurately Reflect Large Language Models' Uncertainty?

Jiayu Liu, Qing Zong, Weiqi Wang, Yangqiu Song
Department of Computer Science and Engineering, HKUST, Hong Kong SAR, China
jliufv@connect.ust.hk; {qzong, wwangbw, yqsong}@cse.ust.hk

Abstract

As Large Language Models (LLMs) are increasingly used in high-stakes domains, accurately assessing their confidence is crucial. Humans typically express confidence through epistemic markers (e.g., "fairly confident") instead of numerical values. However, it remains unclear whether LLMs reliably use these markers to reflect their intrinsic confidence due to the difficulty of quantifying uncertainty associated with various markers. To address this gap, we first define *marker confidence* as the observed accuracy when a model employs an epistemic marker. We evaluate its stability across multiple question-answering datasets in both in-distribution and out-of-distribution settings for open-source and proprietary LLMs. Our results show that while markers generalize well within the same distribution, their confidence is inconsistent in out-of-distribution scenarios. These findings raise significant concerns about the reliability of epistemic markers for confidence estimation, underscoring the need for improved alignment between marker based confidence and actual model uncertainty. Our code is available at https://github.com/HKUST-KnowComp/MarCon.

1 Introduction

LLMs have grown increasingly powerful, yet their application in mission-critical tasks is still hindered by reliability issues (Zhang et al., 2024; Maynez et al., 2020). Therefore, accurately measuring output confidence is crucial for their reliable deployment (Li et al., 2024a; Pedapati et al., 2024; Beigi et al., 2024). Traditionally, black-box confidence estimation in LLMs primarily relies on direct numerical outputs (e.g., "30% confidence") or response consistency (Xiong et al., 2024; Chen and Mueller, 2024; Li et al., 2024b), while white-box methods mainly utilize logits, internal states as information source (Geng et al., 2024). However, natural language is the primary interface for human-LLM interaction. Instead of relying solely on abstract numerical measures, humans often use epistemic markers, such as "I am not sure" or "it is unlikely that," to convey uncertainty (Wallsten, 1986; Erev and Cohen, 1990; Juanchich et al., 2017; Kadavath et al., 2022). This similar recognition of uncertainty markers is essential for effective communication (Willems et al., 2019; Belém et al., 2024), which potentially makes it valuable for LLMs to adopt a similar practice (Yona et al., 2024).

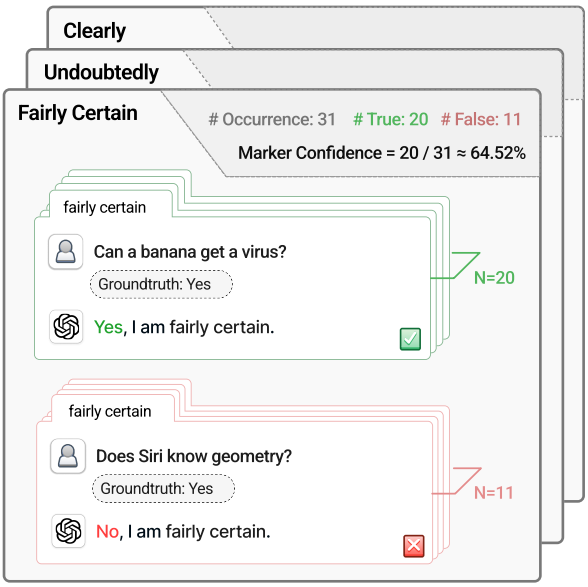


Figure 1: An example of our framework calculating the marker confidence of "fairly certain" for *GPT-4o* on StrategyQA. We calculate the confidence for all the markers across seven models and seven datasets.

However, it remains unexplored whether LLMs are capable of incorporating these epistemic markers in their responses to express their intrinsic confidence stably and consistently. Previous works have primarily concentrated on the misalignment between human and LLM recognition of epistemic markers (Zhou et al., 2024; Tang et al., 2024; Belém et al., 2024), concluding that models always fail to accurately convey confidence in words (Yona

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 206–221
July 27 - August 1, 2025

et al., 2024). In fact, human interpretations of markers are not completely identical (Pennekamp et al., 2024), so even if these markers may not align well with human reasoning, they can still be useful if the model maintains a consistent internal mapping between markers and their actual accuracy. Thus, previous studies questioning the reliability of markers may be insufficient, as they have not examined whether LLMs can consistently apply their own confidence framework.

To address this gap, we investigate whether epistemic markers produced by LLMs reliably reflect their confidence in question-answering tasks. By defining marker confidence as the accuracy of responses when a model uses a specific marker to convey confidence, we calculate the marker confidence with various models and datasets. Additionally, we propose seven evaluation metrics to systematically assess the stability of these markers in both in-distribution and out-of-distribution contexts. Our findings show that while markers perform well within similar distributions, their stability declines in out-of-distribution contexts. Additionally, we compare a range of widely used models and conclude that the more powerful ones demonstrate a better understanding of epistemic markers.

2 Related Work

Our work primarily intersects with confidence estimation in LLMs and studies about epistemic markers. Please find related works in Appendix A.

3 Study Design

3.1 Formalization

Confidence of Epistemic Markers. Let W denote an epistemic marker, $D = \{Q_1, Q_2, \ldots, Q_n\}$ a labeled dataset, and M a model. We define the confidence associated with each epistemic marker as $\mathit{Conf}(W, D, M)$, computed as the accuracy of the answers that explicitly include marker W when the model provides responses. It is important to note that our definition of marker confidence deviates from the conventional interpretation of verbal confidence, which pertains to the semantic uncertainty associated with epistemic markers. To compute the marker confidence for a specific epistemic marker W_i, we use model M_k to generate answers for all questions in the training set of dataset D_j and then extract the subset $Q_{W_i} \subseteq D_j$ consisting of questions whose generated answers contain W_i. The marker confidence is defined as:

$$\mathit{Conf}(W_i, D_j, M_k) = \frac{1}{|Q_{W_i}|} \sum_{q \in Q_{W_i}} \mathbb{I}\Big(M_k(q)\Big),$$

where Q_{W_i} is the set of questions whose generated answers contain the epistemic marker W_i and $\mathbb{I}(\cdot)$ is the indicator function, which is 1 if the answer generated by M_k for question q is correct, and 0 otherwise. An example is provided in Figure 1.

3.2 Methods

We calculate $\mathit{Conf}(W_i, D_j, M_k)$ for all combinations of generated markers, datasets and models (W_i, D_j, M_k) in Appendix B.1 to provide an all-rounded insight into the marker distributions. Specifically, we propose seven metrics to systematically evaluate the stability and consistency of LLM generated epistemic markers:

(1) I-AvgECE *In-domain Average ECE* reflects how well the marker confidence of the model aligns with its actual accuracy in a consistent setting within the same distribution.

(2) C-AvgECE *Cross-domain Average ECE* assesses the calibration error of the marker confidence and the actual accuracy, further reflecting the robustness of the model's marker confidence in out-of-domain scenarios.

(3) NumECE *Numerical* ECE measures the overall calibration of the model's numerical confidence outputs across all datasets. All ECE-related metrics are desired with a lower value, indicating better calibration performance on the target dataset.

(4) MAC *Marker Accuracy Correlation* reflects the correlation between marker confidence and the model accuracy on different datasets. The metric is calculated based on Pearson coefficient, so 0 in this value represents no linear correlation and 1 indicates direct propotional relationship between the marker confidence and model's accuracy.

(5) MRC *Marker Ranking Correlation* measures the model's ability to maintain a consistent marker confidence ranking across different datasets. The metric is calculated based on Spearman coefficient (de Winter et al., 2016), so 0 in this value represents no correlation and 1 indicates totally identical between markers' confidence rankings.

(6) I-AvgCV *In-domain Average CV* captures the dispersion of the model-generated confidence scores within each dataset. A relatively higher I-AvgCV value indicates a more decentralized distribution of markers within the dataset, demonstrating

Model	Marker Confidence					Rank	Density
	I-AvgECE ↓	C-AvgECE ↓	NumECE ↓	C-AvgCV ↓	MAC	MRC ↑	I-AvgCV
Llama-3.1-8B-Instruct	10.09	15.95	22.70	20.80	60.91	11.37	20.48
Qwen2.5-7B-Instruct	7.85	23.60	21.84	31.29	68.06	11.85	22.39
Qwen2.5-14B-Instruct	7.66	20.38	17.98	26.44	73.95	34.60	23.83
Qwen2.5-32B-Instruct	**4.78**	10.40	8.86	19.24	78.20	**36.97**	16.26
Mistral-7B-Instruct-v0.3	10.58	24.81	24.46	28.52	84.57	10.54	21.01
GPT-4o	8.55	**11.84**	**7.56**	**15.72**	76.44	27.54	14.30
GPT-4o-mini	7.65	17.15	12.79	21.98	87.68	16.48	20.61
Average	8.17	17.73	16.60	23.43	75.69	21.34	19.84

Table 1: Model performance across seven metrics. For each metric, the data for the best performing model is bolded. For analytical experiments about markers, we only consider those appear no less than 10 times to eliminate the effect of randomness (see Section 4 for details). All values listed in the table are expressed as percentage (%).

a stronger ability to distinguish between different markers.

(7) C-AvgCV *Cross-domain Average CV* measures the consistency of the model's marker-based confidence across different datasets. A higher C-AvgCV value indicates a greater dispersion of marker confidence across various datasets, suggesting the model's instability regarding marker confidence.

More details about the design and implementation of the metrics can be found in Appendix B.4.1.

4 Experiments and Analysis

Models and Datasets. We experiment with two mainstream open-source and five proprietary LLMs over seven datasets from various domains. More introduction can be found in Appendix B.1.

Baseline. Inspired by previous comparison about using numerical values and uncertainty expression in words to express confidence level (Jaffe-Katz et al., 1989; Knapp et al., 2016), we apply the method of directly prompting the model to express a numerical confidence as baseline for comparison. The prompt designed to elicit epistemic markers and numerical confidence is in Appendix B.1. Our main experiment results are in Table 1.

Marker Filtering Strategies We conduct all marker analysis experiments (namely C-AvgCV, MAC, MRC, and I-AvgCV) by filtering markers that occur fewer than 10 times in the training set in Table 1. The filtering threshold is eventually a tradeoff between the completeness and reliability of the data. On one hand, if the filtering threshold (10 in the main table) is too small, the confidence interval of the results would be large. On the other hand, we cannot include diverse markers for the metrics and get limited insights. The confidence intervals and more details are reported in Appendix B.4.2.

4.1 Main Observations

While the in-distribution marker confidence is relatively stable, it lacks robustness across different datasets. This conclusion is supported by the observation that I-AvgECE values consistently remain lower than NumECE and C-AvgECE values are notably higher than NumECE for 6 out of 7 models, indicating that models exhibit shortcomings in generalizing marker confidence to different distributions and datasets.

More direct evidence to support the conclusion may be inferred from the C-AvgCV of the models. The average C-AvgCV of 0.2343 highlights that marker confidence is highly sensitive to distribution shifts, aligning with the observed high value in C-AvgECE. Notably, we observed that stronger models (e.g., *GPT-4o, Qwen2.5-32B-Instruct*) might exhibit smaller C-AvgCV values.

We further examine the relationship between marker confidence and model accuracy across different datasets using MAC. For 5 out of 7 of the models, the MAC value is over 0.7, which indicates that marker confidence are positively related to the model's accuracy in a strict manner. This also suggests that marker confidence is fragile under distribution shifts, highlighting models' lack of robust understanding of epistemic markers.

Models fail to maintain a consistent ordering of epistemic markers across different domains. The overall low values of MRC suggest that models do not preserve a consistent ranking of markers when applied to datasets with different distributions. We notice that larger models appear to have a better grasp at maintaining a stable ordering of markers. However, both the maximum and average MRC indicates low consistency performance, suggesting a lack of robust understanding of marker

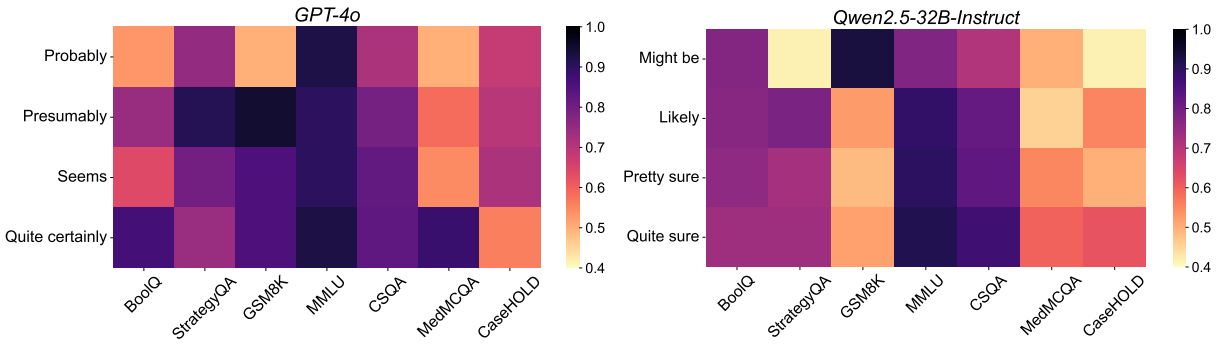


Figure 2: **Model's marker confidence varies greatly across different datasets.** We plot the heatmap of the marker confidence of *GPT-4o* and *Qwen2.5-32B-Instruct* across different datasets, illustrating that even the best models exhibit substantially different confidence levels in various contexts. The markers in the graph are randomly selected from the shared markers of all datasets.

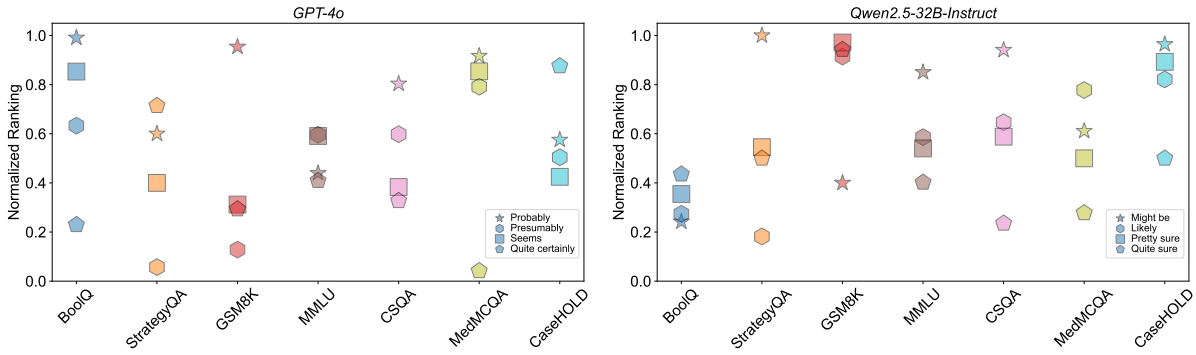


Figure 3: **The rankings of the model's marker confidence fluctuates significantly across different datasets.** We plot the scatter diagram for the marker confidence rankings of the best performing models, but still discovered that the rankings are extremely unstable. The markers in the graph are randomly selected from the shared markers across all datasets.

relative confidence.

The values of the marker confidence are highly concentrated. Since models are expected to express a wide range of confidence including extreme values which is necessary in mission-critical senarios (Alam et al., 2017; Bhise et al., 2018), We expect the models to clearly differentiate the epistemic markers by obtaining a relatively uniform distribution and containing markers with a confidence near 0% or 100%. However, we found the I-AvgCV values typically range from approximately 0.14 and 0.24, demonstrating a concentrated distribution with minor difference. Additionally, only 4 out of 49 settings (dataset, model pair) include markers with confidence under 10% when only those occur no less than 10 times are counted, indicating significant failure in expressing uncertainty.

4.2 Correlation between Performance and Marker Consistency

In Section 4.1, we notice that the statistics in Table 1 suggest that larger models demonstrate a better understanding of epistemic markers, as evidenced by lower C-AvgCV values and higher MRC values. In this section, we quantitatively evaluate the relationship between a model's accuracy and its corresponding C-AvgCV and MRC values to gain a deeper insight into the relationship of model ability and mastery of epistemic markers.

Specifically, for a given model M_k, we use its average accuracy across all datasets as a comprehensive measure of its performance. We then compute the Pearson Correlation Coefficient between each model's overall accuracy and both its C-AvgCV and MRC. The results show a correlation coefficient of -0.88 between model accuracy and C-AvgCV, and a correlation coefficient of 0.75 between model accuracy and MRC. These findings indicate a strong negative relationship between model accuracy and C-AvgCV and a strong positive relationship between model accuracy and MRC. This suggests that more powerful models exhibit greater stability in marker confidence across datasets, as well as a more consistent ordering of markers.

Model	C-AvgCV ↓		MAC		MRC ↑		I-AvgCV	
Threshold	**50**	**100**	**50**	**100**	**50**	**100**	**50**	**100**
Llama-3.1-8B-Instruct	25.50	23.39	77.54	65.98	-8.57	-10.32	11.44	8.59
Qwen2.5-7B-Instruct	30.12	29.30	77.96	69.03	6.01	2.25	18.38	15.41
Qwen2.5-14B-Instruct	26.49	27.66	92.03	94.25	36.56	36.11	20.59	21.13
Qwen2.5-32B-Instruct	20.03	21.27	87.25	86.85	34.91	23.95	13.51	12.18
Mistral-7B-Instruct-v0.3	26.70	26.34	94.80	84.71	30.68	30.60	12.81	9.67
GPT-4o	**15.86**	**16.92**	89.91	90.68	**37.38**	**39.40**	7.52	7.29
GPT-4o-mini	22.15	22.41	86.82	87.45	24.16	24.36	11.98	11.39
Average	23.84	23.90	86.62	82.71	23.02	20.91	13.75	12.24

Table 2: This table presents the results (in %) of the marker analysis experiments, organized by different filtering thresholds. We observed that the conclusion obtained from Table 1 still holds when the threshold increases.

4.3 Conclusion Robustness under Different Filtering Thresholds

Our primary conclusions regarding marker consistency and model understanding of epistemic markers remain robust across various filtering thresholds. As detailed in Table 2, even when increasing the filtering threshold to 50 or 100 occurrences, the observed trends persist: 1) The C-AvgCV and MRC values consistently remain low, while the I-AvgCV values remain high. This pattern continues to suggest that models struggl to maintain consistent confidence across different datasets and to differentiate effectively between markers. 2) The MAC values consistently remain high. This reinforces the strong positive relationship between model performance and its marker usage ability. Moreover, **the observed shortcomings extend even to frequently occurring markers** (with over 100 instances), indicating a severe challenge in the model's ability to reliably utilize epistemic expressions. This pervasive issue further substantiates our claims regarding the difficulties in ensuring reliable marker usage.

4.4 Discussions

This section aims to explain the observed differential in the consistent deployment of epistemic markers within in-domain versus out-of-domain contexts. Furthermore, we aim to investigate the distributional characteristics that potentially modulate the confidence associated with these markers.

Within similar distributions, models tend to maintain a stable "preference" for employing these markers to express uncertainty. For instance, in in-domain scenarios, responses on the test set leverage a consistent pattern of epistemic marker "preference" due to the similarity of the distribution. However, this consistency breaks down when the data distribution shifts. As the distribution changes, the model's preference usage for epistemic markers also changes, hindering the transfer of marker-based confidence to test sets from different datasets.

To further investigate this phenomenon, we conducted an in-depth analysis by calculating the average ECE values for each dataset to quantify the generalizability of the marker consistency from the training set to the test set. This involved averaging the ECE values across seven models in the in-domain scenario. Our analysis revealed that **datasets with greater component diversity generally exhibit a higher ECE**. This suggests that more complex or multi-sourced distributions make it challenging to transfer marker confidence from the training to the test set, as exemplified by datasets like MMLU and StrategyQA, which are not domain-specific. This observation further supports our claim that models perform well in in-domain scenarios primarily due to the consistency of the underlying data distribution. The details about the experiments are reported in Appendix C.

5 Conclusion

Our study evaluates whether LLMs can reliably express confidence using epistemic markers. We define marker confidence as the observed accuracy of responses containing specific markers, conduct extensive experiments and evaluate the results with several metrics. The results show that the marker confidence shifts significantly under distribution changes, following the trend of model accuracy, which highlights poor stability. Additionally, models struggle to effectively differentiate between markers and maintain consistent marker rankings across datasets. These findings suggest that the LLM generated markers to express their confidence is unreliable and requires improved alignment between verbal confidence and actual performance. Our work contributes to more consistent confidence estimation frameworks, ultimately facilitating reliable and trustworthy LLM responses.

Limitation

Human language is remarkable for its complexity, variability, and rich connotations, particularly when expressing uncertainty. Within a complete linguistic system, factors like sentence structure can significantly influence confidence, which is often difficult to quantify with epistemic markers alone. Moreover, in the context of long-form communication, it is clear that confidence cannot be simply measured by the confidence values of epistemic markers. To facilitate simplicity in evaluation and to focus on the study of epistemic markers, we adopt a relatively idealized approach: using epistemic markers generated by LLMs in closed-source QA tasks to represent the confidence of the responses while keeping them relatively brief. Additionally, epistemic markers may carry different meanings across cultures and languages. However, we only consider epistemic markers in English.

Despite our idealized conditions and using state-of-the-art models, LLMs still fail to consistently align epistemic markers with their true confidence levels, revealing that the issue lies not only with our approach but also with the models themselves. While they perform well in question-answering tasks, they do not truly understand epistemic markers (Zhou et al., 2023), struggle to express consistent confidence in these markers, and have difficulty aligning their confidence expressions with human expectations (Belém et al., 2024). This points to a deeper challenge in model behavior, suggesting that future research should focus on addressing fundamental gaps in model linguistic alignment.

Ethics Statement

Our work contains no offensive contents and have no potential risks.

Licenses. We will share our code under the MIT license, allowing other researchers free access to our resources for research purposes. The datasets used in this paper, including BoolQ (Clark et al., 2019), StrategyQA (Geva et al., 2021), GSM8K (Cobbe et al., 2021), MMLU (Hendrycks et al., 2021), CSQA (Talmor et al., 2019), MedMCQA (Pal et al., 2022), and CaseHOLD (Zheng et al., 2021), are shared under either the CC BY-SA license, Apache License Version 2.0, or the MIT License, all of which permit their use for research purposes. As for language models, we access all open-source LMs via the Huggingface Hub (Wolf et al., 2020). Our use of *GPT-4o* (OpenAI, 2024b) and *GPT-4o-mini* (OpenAI, 2024a) is conducted through OpenAI's official website[1]. All associated licenses permit user access for research purposes, and we have agreed to adhere to all terms of use.

Acknowledgements

We thank the anonymous reviewers and the area chair for their constructive comments. The authors of this paper were supported by the ITSP Platform Research Project (ITS/189/23FP) from ITC of Hong Kong, SAR, China, and the AoE (AoE/E-601/24-N), the RIF (R6021-20) and the GRF (16205322) from RGC of Hong Kong,SAR, China.

References

Rahul Alam, Sudeh Cheraghi-Sohi, Maria Panagioti, Aneez Esmail, Stephen Campbell, and Efharis Panagopoulou. 2017. Managing diagnostic uncertainty in primary care: a systematic critical review. *BMC Family Practice*, 18:1–13.

Mohammad Beigi, Ying Shen, Runing Yang, Zihao Lin, Qifan Wang, Ankith Mohan, Jianfeng He, Ming Jin, Chang-Tien Lu, and Lifu Huang. 2024. Internalinspector i^2: Robust confidence estimation in llms through internal states. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 12847–12865. Association for Computational Linguistics.

Catarina G. Belém, Markelle Kelly, Mark Steyvers, Sameer Singh, and Padhraic Smyth. 2024. Perceptions of linguistic uncertainty by language models and humans. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 8467–8502. Association for Computational Linguistics.

Viraj Bhise, Suja S Rajan, Dean F Sittig, Robert O Morgan, Pooja Chaudhary, and Hardeep Singh. 2018. Defining and measuring diagnostic uncertainty in medicine: a systematic review. *Journal of general internal medicine*, 33:103–115.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish,

[1]https://platform.openai.com/docs/api-reference/introduction

Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Guiming Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or llms as the judge? A study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 8301–8327. Association for Computational Linguistics.

Jiuhai Chen and Jonas Mueller. 2024. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 5186–5200. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Joost C. F. de Winter, Samuel D. Gosling, and Jeff Potter. 2016. Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychological Methods*, 21(3):273–290.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 5050–5063. Association for Computational Linguistics.

Ido Erev and Brent L Cohen. 1990. Verbal versus numerical probabilities: Efficiency, biases, and the preference paradox. *Organizational Behavior and Human Decision Processes*, 45(1):1–18.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Trans. Assoc. Comput. Linguistics*, 8:539–555.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 6577–6595. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Trans. Assoc. Comput. Linguistics*, 9:346–361.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Mengting Hu, Zhen Zhang, Shiwan Zhao, Minlie Huang, and Bingzhe Wu. 2023. Uncertainty in natural language processing: Sources, quantification, and applications. *CoRR*, abs/2306.04459.

Amanda Jaffe-Katz, David V Budescu, and Thomas S Wallsten. 1989. Timed magnitude comparisons of numerical and nonnumerical expressions of uncertainty. *Memory & Cognition*, 17(3):249–264.

Zahra Jalilibal, Amirhossein Amiri, Philippe Castagliola, and Michael B. C. Khoo. 2021. Monitoring the coefficient of variation: A literature review. *Comput. Ind. Eng.*, 161:107600.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Marie Juanchich, Amélie Gourdon-Kanhukamwe, and Miroslav Sirota. 2017. "i am uncertain" vs "it is uncertain". how linguistic markers of the uncertainty source affect uncertainty communication. *Judgment and Decision Making*, 12(5):445–465.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *CoRR*, abs/2207.05221.

Peter Knapp, Peter H Gardner, and Elizabeth Woolf. 2016. Combined verbal and numerical expressions increase perceived risk of medicine side-effects: a randomized controlled trial of ema recommendations. *Health Expectations*, 19(2):264–274.

Dongryeol Lee, Yerin Hwang, Yongil Kim, Joonsuk Park, and Kyomin Jung. 2024. Are llm-judges robust to expressions of uncertainty? investigating the effect of epistemic markers on llm-based evaluation. *CoRR*, abs/2410.20774.

Moxin Li, Wenjie Wang, Fuli Feng, Fengbin Zhu, Qifan Wang, and Tat-Seng Chua. 2024a. Think twice before assure: Confidence estimation for large language models through reflection on multiple answers. *CoRR*, abs/2403.09972.

Moxin Li, Wenjie Wang, Fuli Feng, Fengbin Zhu, Qifan Wang, and Tat-Seng Chua. 2024b. Think twice before trusting: Self-detection for large language models through comprehensive answer reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 11858–11875. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Trans. Mach. Learn. Res.*, 2022.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with confidence: Uncertainty quantification for black-box large language models. *Trans. Mach. Learn. Res.*, 2024.

Jiayu Liu, Junhao Tang, Hanwen Wang, Baixuan Xu, Haochen Shi, Weiqi Wang, and Yangqiu Song. 2024. Gprooft: A multi-dimension multi-round fact checking framework based on claim fact extraction. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 118–129.

Huan Ma, Changqing Zhang, Yatao Bian, Lemao Liu, Zhirui Zhang, Peilin Zhao, Shu Zhang, Huazhu Fu, Qinghua Hu, and Bingzhe Wu. 2023. Fairness-guided few-shot prompting for large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan T. McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1906–1919. Association for Computational Linguistics.

OpenAI. 2024a. Gpt-4o mini: advancing cost-efficient intelligence. *OpenAI*.

OpenAI. 2024b. Hello gpt-4o. *OpenAI*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning, CHIL 2022, 7-8 April 2022, Virtual Event*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Tejaswini Pedapati, Amit Dhurandhar, Soumya Ghosh, Soham Dan, and Prasanna Sattigeri. 2024. Large language model confidence estimation via black-box access. *CoRR*, abs/2406.04370.

Pia Pennekamp, Jamal K. Mansour, and Rhiannon J. Batstone. 2024. Variability in verbal eyewitness confidence. *Applied Cognitive Psychology*.

Limin Su and Huimin Li. 2021. Project procurement method decision-making with spearman rank correlation coefficient under uncertainty circumstances. *Int. J. Decis. Support Syst. Technol.*, 13(2):16–44.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4149–4158. Association for Computational Linguistics.

Zhisheng Tang, Ke Shen, and Mayank Kejriwal. 2024. An evaluation of estimative uncertainty in large language models. *arXiv preprint arXiv:2405.15185*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Akim Tsvigun, Daniil Vasilev, Rui Xing, Abdelrahman Boda Sadallah, Lyudmila Rvanova, Sergey Petrakov, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, Maxim Panov, and Artem Shelmanov. 2024. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *CoRR*, abs/2406.15627.

Thomas S Wallsten. 1986. Measuring the vague meanings of probability terms. *Journal of Experimental Psychology: General*, 115(4):348–365.

Weiqi Wang, Limeng Cui, Xin Liu, Sreyashi Nag, Wenju Xu, Chen Luo, Sheikh Muhammad Sarwar, Yang Li, Hansu Gu, Hui Liu, Changlong Yu, Jiaxin Bai, Yifan Gao, Haiyang Zhang, Qi He, Shuiwang Ji, and Yangqiu Song. 2025. Ecomscriptbench: A multi-task benchmark for e-commerce script planning via step-wise intention-driven product association. *Preprint*, arXiv:2505.15196.

Weiqi Wang, Tianqing Fang, Wenxuan Ding, Baixuan Xu, Xin Liu, Yangqiu Song, and Antoine Bosselut. 2023a. CAR: conceptualization-augmented reasoner for zero-shot commonsense question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 13520–13545. Association for Computational Linguistics.

Weiqi Wang, Tianqing Fang, Chunyang Li, Haochen Shi, Wenxuan Ding, Baixuan Xu, Zhaowei Wang, Jiaxin Bai, Xin Liu, Cheng Jiayang, Chunkit Chan, and Yangqiu Song. 2024a. CANDLE: iterative conceptualization and instantiation distillation from large language models for commonsense reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 2351–2374. Association for Computational Linguistics.

Weiqi Wang, Tianqing Fang, Haochen Shi, Baixuan Xu, Wenxuan Ding, Liyu Zhang, Wei Fan, Jiaxin Bai, Haoran Li, Xin Liu, and Yangqiu Song. 2024b. On the role of entity and event level conceptualization in generalizable reasoning: A survey of tasks, methods, applications, and future directions. *CoRR*, abs/2406.10885.

Weiqi Wang, Tianqing Fang, Baixuan Xu, Chun Yi Louis Bo, Yangqiu Song, and Lei Chen. 2023b. CAT: A contextualized conceptualization and instantiation framework for commonsense reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13111–13140. Association for Computational Linguistics.

Weiqi Wang and Yangqiu Song. 2024. MARS: benchmarking the metaphysical reasoning abilities of language models with a multi-task evaluation dataset. *CoRR*, abs/2406.02106.

Sanne J. W. Willems, Casper J. Albers, and Ionica Smeets. 2019. Variability in the interpretation of dutch probability phrases - a risk for miscommunication. *Preprint*, arXiv:1901.09686.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Hui Xiong, Shashi Shekhar, Pang-Ning Tan, and Vipin Kumar. 2004. Exploiting a support-based upper bound of pearson's correlation coefficient for efficiently identifying strongly correlated pairs. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 334–343. ACM.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V. Chawla, and Xiangliang Zhang. 2024. Justice or prejudice? quantifying biases in llm-as-a-judge. *CoRR*, abs/2410.02736.

Gal Yona, Roee Aharoni, and Mor Geva. 2024. Can large language models faithfully express their intrinsic uncertainty in words? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 7752–7764. Association for Computational Linguistics.

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng. 2024. Self-alignment for factuality: Mitigating hallucinations in llms via self-evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 1946–1965. Association for Computational Linguistics.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. 2021. When does

pretraining help?: assessing self-supervised learning for law and the casehold dataset of 53, 000+ legal holdings. In *ICAIL '21: Eighteenth International Conference for Artificial Intelligence and Law, São Paulo Brazil, June 21 - 25, 2021*, pages 159–168. ACM.

Kaitlyn Zhou, Jena D. Hwang, Xiang Ren, and Maarten Sap. 2024. Relying on the unreliable: The impact of language models' reluctance to express uncertainty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3623–3643. Association for Computational Linguistics.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. 2023. Navigating the grey area: How expressions of uncertainty and overconfidence affect language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5506–5524. Association for Computational Linguistics.

Qing Zong, Zhaowei Wang, Tianshi Zheng, Xiyu Ren, and Yangqiu Song. 2024. Comparisonqa: Evaluating factuality robustness of llms through knowledge frequency control and uncertainty. *CoRR*, abs/2412.20251.

Appendices

A Related Work

Confidence Estimation in LLMs. Confidence estimation in LLMs refers to the process of assessing a model's confidence in its output. Previous research on this topic can be categorized into white-box and black-box methods, distinguished by whether they utilize the model's internal information. White-box methods leverage the internal states of LLMs, with key approaches including information-based methods that analyze these inner states (Vashurin et al., 2024; Geng et al., 2024; Burns et al., 2023), such as perplexity (Fomicheva et al., 2020; Zong et al., 2024), the negative log probability of generated tokens (Duan et al., 2024), and others. In the field of black-box methods, Lin et al. (2022) first introduces the concept *verbal confidence* that prompts LLM to output its confidence directly. Most subsequent methods are based on either directly prompting the model to generate an output or consistency sampling (Lin et al., 2024; Xiong et al., 2024; Chen and Mueller, 2024; Liu et al., 2024). However, previous methods primarily focus on processing numerical values to estimate the LLM's confidence, leading to a research gap in exploring LLM confidence expression through linguistic patterns, especially epistemic markers.

Studies on Epistemic Markers. Epistemic markers are essential for expressing confidence in conversation, playing a key role in human-LLM interactions (Hu et al., 2023). Recent studies have examined the reliability of LLMs in mastering epistemic markers, primarily investigating their interpretation of various uncertainty expressions. For instance, Tang et al. (2024) and Belém et al. (2024) use sentence templates with uncertainty expressions to prompt the model to assess overall confidence, though this approach is limited by the fixed nature of the templates, restricting generalizability. Zhou et al. (2023) and Zhou et al. (2024) argue that LLMs often mimic marker distributions from training data rather than truly understanding them, with the latter highlighting a tendency for overconfidence. Lee et al. (2024) also examine epistemic markers but focus on robustness and biases in model interpretations. While these studies investigate LLM generation of epistemic markers, our work aligns most closely with Yona et al. (2024), which directly challenges the ability of LLMs to accurately convey confidence using epistemic mark-

ers. However, they employ LLM-as-a-judge and few-shot prompting to assess the numerical confidence of uncertainty expressions, introducing potential bias (Chen et al., 2024; Ye et al., 2024; Ma et al., 2023). Additionally, they use human judges to assess the quality of LLM judges, which essentially aligns the model's interpretation of markers with human understanding. This approach is similar to other works in the field, which also focus on aligning human and LLM recognition of epistemic markers. However, as long as LLMs maintain a consistent framework for incorporating epistemic markers to express confidence, we can learn from its intrinsic mapping of marker usage and align it with human expectations through external means.

B Technical Details

B.1 Experiment Setup

Models We incorporate a range of commonly used LLMs of different scales, including Llama-3.1-8B-Instruct (Touvron et al., 2023), Qwen2.5-7B-Instruct, Qwen2.5-14B-Instruct, Qwen2.5-32B-Instruct (Yang et al., 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), GPT-4o (OpenAI, 2024b), GPT-4o-mini (OpenAI, 2024a). For all models, we use a temperature of 0.5 to balance between logical consistency and creativity. All the open-source models are run on 4 NVIDIA A6000 (40G) GPUs with BF16.

We observed that instruction-tuned models exhibit greater variation and demonstrate better linguistic diversity when using epistemic markers. Although we also tested some base models, we found that for the same dataset, they emitted significantly fewer markers compared to the instruction-tuned models (See Figure 4) . As a result, we chose to focus our experiments on instruction-tuned models instead.

Datasets We benchmark the LLMs' responses with confidence expression using epistemic markers using the following seven datasets requiring knowledge in different domains: 1) Factual and commonsense knowledge: BoolQ (Clark et al., 2019), StrategyQA (Geva et al., 2021), CSQA (Talmor et al., 2019). 2) Mathematical reasoning: GSM8K (Cobbe et al., 2021). 3) Medical reasoning: MedMCQA (Pal et al., 2022). 4) Law reasoning: CaseHOLD (Zheng et al., 2021). 5) Mixed factual datasets: MMLU (Hendrycks et al., 2021).

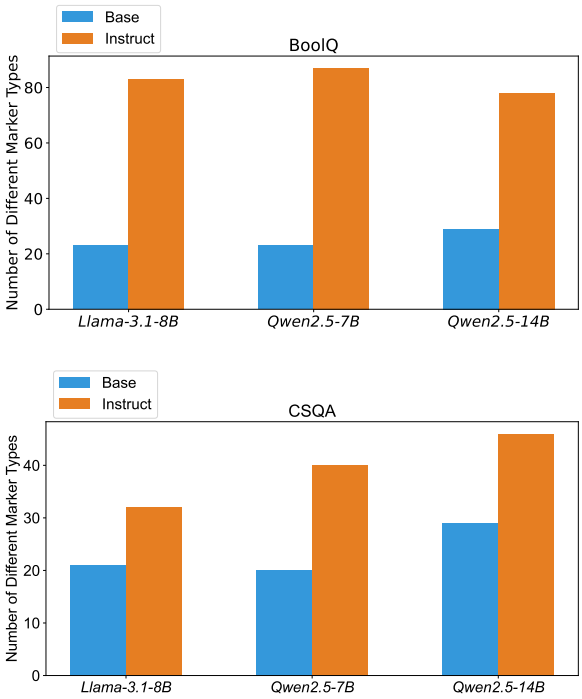


Figure 4: The number of epistemic markers that six different models generated in BoolQ and CSQA dataset. The results indicate that the instruct-tuned models exhibit much better linguistic diversity than base models in expressing confidence, which is desired by our experiment.

Prompts The prompts used in our experiments are shown in Table 4. We performed five permutations on the prompt with *Qwen2.5-7B-Instruct* and *Mistral-7B-Instruct-v0.3* on BoolQ, StrategyQA, CSQA, MedMCQA and found that both accuracy and C-AvgCV, MAC, MRC and I-AvgCV values varied only slightly. Consequently, we randomly selected one version and conducted all subsequent experiments using it.

B.2 Model Sources

This section clarifies the sources of the models used in our study. For methods involving LLMs, we utilize their instruction fine-tuned versions (see Appendix B.1 for more details) accessed via the Hugging Face Hub (Wolf et al., 2020). Specifically, for *Llama-3.1-8B-Instruct*, we employ the version *meta-llama/Llama-3.1-8B-Instruct*. The models related to Qwen include *Qwen/Qwen2.5-7B-Instruct*, *Qwen/Qwen2.5-14B-Instruct*, and *Qwen/Qwen2.5-32B-Instruct*. For *Mistral-7B-Instruct-v0.3*, we use *mistralai/Mistral-7B-Instruct-v0.3*.

B.3 Detailed Implementation

Data Preprocessing We applied data preprocessing methods to the datasets. For GSM8K, we transformed it into a binary-question dataset for convenience. Specifically, for half of the questions Q_i with even index in GSM8K, we first extracted the correct answer A_i, then used a question template to create a binary question by incorporating (Q_i, A_i) and setting the correct binary answer to "yes." For the remaining half of the questions Q_j, we randomly selected an answer A_j different from the correct answer, and used the same question template to create a binary question by incorporating (Q_j, A_j) and setting the correct binary answer to "no." The question template and two examples are given in Table 5. For the training set of the MMLU, due to its massive size, we randomly sampled a subset of 20000 QA-pairs for the MMLU training set. For MedMCQA dataset, for the convenience of evaluation, we pick the subset of the answer with only one correct answer. We then randomly sample 9686 QA-pairs for the MedMCQA training set and 2422 for its test set. For BoolQ, we did not expose the model to the "passage" part and treated it as a closed-book question-answering dataset in our experiment. Since CaseHOLD isn't explicit split into training set and test set, we divide the former 80% as training set and the rest as test set. A detailed statistics for our dataset usage is on Table 3.

Dataset	Train Size	Test Size
BoolQ	9427	3270
StrategyQA	2061	229
GSM8K	7473	1319
MMLU	20000	14041
CSQA	9741	1221
MedMCQA	9686	2422
CaseHOLD	8396	2099

Table 3: A detailed statistcs of our dataset usage.

Epistemic Marker Extraction For each model, we extract the epistemic marker from each response by few-shot prompting (Brown et al., 2020) the same model to recognize the epistemic markers emitted by itself. We manually examined a subset of each dataset and find out most of them are able to recognize the epistemic markers. For the unrecognized ones or the one that didn't match the desired format, we uniformly use *GPT-4o-mini* to extract its epistemic markers. According to Zhou et al. (2024), models are relunctant to express confidence in words, so we also provided responses that did not include any epistemic markers as few-shot samples, and those with no markers are grouped together as a special epistemic marker.

B.4 Evaluation Metrics

This section introduces the evaluation metrics in detail and explain our experiment settings.

B.4.1 Detailed Implementation and Formulas

Our evaluation metrics are categorized into three kinds: ECE-based (Xiong et al., 2024), CV-based (Jalilibal et al., 2021) and Spearman/Pearson coefficient based (Su and Li, 2021; Xiong et al., 2004), aiming to reflect the calibration error, the degree of dispersion and correlation respectively.

NumECE, I-AvgECE, and C-AvgECE To evaluate the calibration of model-generated confidence, we introduce three metrics: NumECE, I-AvgECE, and C-AvgECE. Since the latter two are based on generalization of marker confidence, these metrics assess the model's stability on marker confidence, considering both within-domain (I-AvgECE) and cross-domain (C-AvgECE) scenarios and comparing with number-based methods.

NumECE measures the overall calibration of the model's outputted numerical confidence. For each dataset D_j, we compute the expected calibration error (ECE-num) based on the model's numerical confidence on the test set. The final NumECE is the average of these ECE values across all datasets, providing a comprehensive evaluation of the model's confidence calibration.

I-AvgECE focuses on the model's performance within the same domain, where the training and test datasets are identical. For each dataset D_p, we calculate the marker-based expected calibration error (ECE-mar) by using the marker's confidence $\mathit{Conf}(W_i, D_p, M_k)$ obtained from the training set of D_p on the test set of it when the model also emits W_i as confidence expression. The final I-AvgECE is obtained by averaging these values across all datasets as well.

C-AvgECE evaluates the model's ability to generalize its marker confidence across different datasets. For each pair of distinct datasets (D_p, D_q), where $D_p \neq D_q$, we calculate the marker-based expected calibration error ECE-mar(D_p, D_q, M_k). This is done by using the

	Binary Question	Multiple Choice Question
Eliciting epistemic markers	*User*: The following is a binary question. When responding, answer with a binary answer from **[choices]** and incorporate only one epistemic marker to reflect your confidence level. You must include your binary answer at the beginning of your response then respond with the epistemic markers in a concise and brief manner. The question is: **[Question]** And your answer is:	*User*: The following is a multiple choice question. When responding, answer with a letter from **[choices]** and incorporate only one epistemic marker to reflect your confidence level. You must include your choice of letter at the beginning of your response then respond with the epistemic markers in a concise and brief manner. The question is: **[Question]** The options are: **[Options]** And your answer is:
Eliciting numerical values	*User*: The following is a binary question. When responding, answer with a binary answer from **[choices]** and incorporate a number between 0 and 100 to reflect your confidence level. You must include your binary answer at the beginning of your response then respond with the confidence score in a concise and brief manner. The question is: **[Question]** And your answer is:	*User*: The following is a multiple choice question. When responding, answer with a letter from **[choices]** and incorporate a number between 0 and 100 to reflect your confidence level. You must include your choice of letter at the beginning of your response then respond with the confidence score in a concise and brief manner. The question is: **[Question]** The options are: **[Options]** And your answer is:

Table 4: Our prompt to elicit epistemic markers and numerical confidence values. The text inside the square brackets is filled with actual content in the dataset. Specifically, choices are capital letters that represent the options (e.g., "A, B, C, D" or "A, B, C, D, E") for multiple choice questions and "yes or no" for binary questions.

model's confidence *Conf*(W_i, D_p, M_k) on the training dataset D_p to estimate the model's marker confidence on the test set of D_q, thereby measuring the model's ability to transfer its marker confidence to a new dataset. The final C-AvgECE is computed by averaging the ECE-mar values across all dataset pairs, providing insight into the model's robustness in handling cross-distribution variations. The mathematical formulas of three ECE-based metrics is given by:

$$\text{NumECE} = \frac{1}{|D|} \sum_{D_j \in D} \text{ECE-num}(D_j, M_k),$$

$$\text{I-AvgECE} = \frac{1}{|D|} \sum_{D_p \in D} \text{ECE-mar}(D_p, D_p, M_k),$$

$$\text{C-AvgECE} = \frac{\sum_{\substack{D_p, D_q \in D \\ D_p \neq D_q}} \text{ECE-mar}(D_p, D_q, M_k)}{|D| * (|D| - 1)}$$

where $|D|$ is the total number of datasets $(D_1, D_2, \ldots, D_n)$. Note that for all ECE values, we use a ECE bin number of *N*, where *N* is the number of confidence predictions.

I-AvgCV and C-AvgCV To quantify the concentration and variation of marker confidence, we propose I-AvgCV and C-AvgCV. These metrics assess how dispersed and consistent marker confidence is within individual datasets and across datasets, respectively.

I-AvgCV measures the concentration of marker confidence within a single dataset. For each dataset D_j, we calculate the coefficient of variation (CV) of the confidence of different markers *Conf*(W_i, D_j, M_k). The final I-AvgCV is the average CV value across all datasets, providing an overall measure of confidence concentration.

It is important to note that while we expect the

Question Template
For the question **[original question]**, is the answer **[original answer]** its correct answer?
Sample 1
Q_i: Each bird eats 12 beetles per day, each snake eats 3 birds per day, and each jaguar eats 5 snakes per day. If there are 6 jaguars in a forest, how many beetles are eaten each day? A_i: 1080 (The correct answer is 1080) Integrated binary question: For the question 'Each bird eats 12 beetles per day, each snake eats 3 birds per day, and each jaguar eats 5 snakes per day. If there are 6 jaguars in a forest, how many beetles are eaten each day?", is the answer 1080 its correct answer? Binary answer: Yes.
Sample 2
Q_j: James writes a 3 - page letter to 2 different friends twice a week. How many pages does he write a year? A_j: 223 (Randomly generated answer, the correct answer is 624) Integrated binary question: For the question 'James writes a 3 - page letter to 2 different friends twice a week. How many pages does he write a year?", is the answer 223 its correct answer? Binary answer: No.

Table 5: Data pre-processing method used in GSM8K. The text inside the square brackets is replaced by actual content in the dataset. Sample 1 keeps the original correct answer and incorporate it into the binary answer while setting the binary answer to "Yes." Sample 2 randomly generates a wrong answer and set the binary answer to "No."

distribution of marker confidence to be more dispersed, we are not claiming that greater dispersion is inherently better. Our desired result for models is to cover a relatively wide range of confidence values across all markers, while also clearly differentiating between different markers. This would facilitate more effective confidence expression in a variety of scenarios. However, as shown in Table 1, the average I-AvgCV value marker is lower than 0.2, which indicates that the marker confidence is highly concentrated, leading us to conclude that the model fails to clearly differentiate between the markers.

C-AvgCV evaluates the consistency of marker confidence across datasets. For each shared marker (markers that appear in each dataset) W_i, we compute the CV of the marker confidence across different datasets and then average these values over all shared markers. The final C-AvgCV quantifies the consistency of model-generated confidence across multiple datasets.

The mathematical formulations for I-AvgCV and C-AvgCV are given by:

$$CV(D_j, M_k) = \frac{\sigma(D_j, M_k)}{\mu(D_j, M_k)},$$

$$\text{I-AvgCV}(M_k) = \frac{1}{|D|} \sum_{j=1}^{|D|} CV(D_j, M_k),$$

$$CV(W_i, M_k) = \frac{\sigma(W_i, M_k)}{\mu(W_i, M_k)},$$

$$\text{C-AvgCV}(M_k) = \frac{1}{|W|} \sum_{i=1}^{|W|} CV(W_i, M_k)$$

where $|W|$ is the number of shared markers across every datasets for model M_k, $\sigma(D_j, M_k)$ is the standard deviation of the confidence scores for all markers in dataset D_j for model M_k, $\mu(D_j, M_k)$ is the mean of the confidence scores for all markers in dataset D_j for model M_k, $\sigma(W_i, M_k)$ is the standard deviation of the confidence scores for the marker W_i across different datasets for model M_k and $\mu(W_i, M_k)$ is the mean of the confidence scores for the marker W_i across different datasets for model M_k.

MRC To assess the alignment of marker rankings across datasets, we introduce a metric based on the Spearman rank correlation coefficient: *Marker Rank Correlation* (MRC). This metric capture the degree of consistency in marker confidence rankings across datasets.

Specifically, For each pair of datasets (D_p, D_q), we compute the Spearman rank correlation coefficient between the confidence rankings of shared markers. The final MRC for the model is the average correlation across all dataset pairs. The mathematical formulations for MRC are given by:

$$\text{MRC} = \frac{1}{|P|} \sum_{\substack{(D_p, D_q) \in P \\ D_p \neq D_q}} \rho(D_p, D_q)$$

where $W_1, \ldots, W_i$ are all the epistemic markers that shared by D_p and D_q, $\text{S}(X, Y)$ denotes the Spearman rank correlation coefficient between the rankings of X and Y and $\text{Conf}(W_i, D_j, M_k)$ represents the confidence of marker W_i for model M_k on dataset D_j.

MAC To analyze whether the confidence of markers and the accuracy of the model are positively correlated, we propose the *Marker Accuracy Correlation* (MAC) based on the Pearson correlation coefficient.

Specifically, for a given model M_k, we consider the confidence of a specific shared marker W_i, which is present across all datasets associated with M_k. We then compute the Pearson correlation coefficient between the set of marker confidences across these datasets and the model's overall accuracies on the same datasets. Finally, we compute the average of the correlation coefficients $\rho(W_i, M_k)$ across all shared markers W_i to obtain the overall correlation coefficient for the model, denoted as $\text{MAC}(M_k)$. It's mathematical formula is given by:

$$\text{MAC}(M_k) = \frac{1}{|W|} \sum_{W_i \in W} \rho(W_i, M_k),$$

where W is the set of all shared markers W_i, $|W|$ is the number of all shared markers and $\rho(W_i, M_k)$ is the Pearson correlation coefficient between the confidence of marker W_i and the model's accuracy on all the datasets.

These metrics provide a quantitative assessment of the consistency and concentration of model-generated confidence values across different datasets.

Threshold	Confidence Interval
10	∼23%
50	∼10%
200	∼5%

Table 6: The exact confidence interval of different filtering thresholds.

B.4.2 Details on Marker Filtering Strategies

The filtering is necessary because our method of quantifying marker confidence is based on accuracy, which is affected by the confidence interval. If the sample size for a particular marker is too small, its corresponding confidence values can be heavily influenced by random variations. For instance, if the marker "unsure" appears only once in the training set and the response happens to be correct, the marker confidence for "unsure" would be 100%, which may not accurately reflect the model's true intent. Previous works also show that confidence expression could be more reflective for humans when the it is determined in a crowd-source manner (Pennekamp et al., 2024), which supports our setting.

Furthermore, we have enough shared markers (more than ten for each model) after implementing the filtering, which ensures the reliability of our experiment.

On the other hand, all epistemic markers obtained from the training set are used for the experiment related to ECE values. Since the estimated confidence for each question in the test set is derived from the marker confidence in the training set, it is essential to ensure that the vast majority of markers in the test set can be mapped to corresponding markers in the training set. This approach is reasonable since the low frequency of marker occurrences results in minimal impact on the overall calibration performance, which ensures both completeness and reliability.

C Distribution Analysis

Our investigation primarily focused on the influence of **distribution diversity** and **difficulty** on the generalizability of marker confidence. We measured this generalizability by calculating the Average ECE value across various models for each specific dataset.

Table 7 illustrates that multi-domain datasets, specifically StrategyQA (14.42%) and MMLU (15.97%), exhibit significantly higher Average ECE

values compared to single-domain datasets such as GSM8K (Math, 6.68%), MedMCQA (Medical, 6.25%), and CaseHOLD (Law, 9.53%). This finding leads to the conclusion that a more diverse data distribution impedes the transfer of models' marker confidence preferences from the training to the test set (Wang et al., 2023b,a, 2024a,b, 2025; Wang and Song, 2024).

Dataset	Domain	Average ECE
StrategyQA	Multi-domain	14.42
MMLU	Multi-domain	15.97
GSM8K	Math	6.68
MedMCQA	Medical	6.25
CaseHOLD	Law	9.53

Table 7: The Average ECE values across different models for certain dataset. We found that multi-domain datasets exhibit higher average ECE values than single-domain ones.

Additionally, we observed that a substantial **difficulty gap** between the training and test sets of the marker confidence also compromises its transferability. As shown in Table 8, we utilized a challenging math dataset, MATH-500, and a simpler math dataset, GSM8K, to compute the MRC value, representing the relevance of marker rankings across these two datasets. For both *Qwen2.5-7B-Instruct* (0.38) and *Mistral-7B-Instruct-v0.3* (0.26), the models demonstrated low relevance in marker rankings obtained from datasets with differing difficulties. This indicates that the **difficulty of the data distribution** can significantly affect the generalizability of marker confidence preferences.

Model	MRC
Qwen2.5-7B-Instruct	0.38
Mistral-7B-Instruct-v0.3	0.26

Table 8: The MRC value for two models across MATH-500 and GSM8K. The results shows that marker rankings obtained from the two datasets are relevant to little extent.

Limited-Resource Adapters Are Regularizers, Not Linguists

Marcell Fekete[1] **Nathaniel R. Robinson**[2] **Ernests Lavrinovics**[1]
E. Djeride Jean-Baptiste[3] **Raj Dabre**[4] **Johannes Bjerva**[1] **Heather Lent**[1]
[1]Aalborg University, Denmark; [2]Johns Hopkins University, USA;
[3]University of Ottawa, Canada; [4]IIT Madras, India
{mrfe, hcle}@cs.aau.dk

Abstract

Cross-lingual transfer from related high-resource languages is a well-established strategy to enhance low-resource language technologies. Prior work has shown that adapters show promise for, e.g., improving low-resource machine translation (MT). In this work, we investigate an adapter souping method combined with cross-attention fine-tuning of a pre-trained MT model to leverage language transfer for three low-resource Creole languages, which exhibit relatedness to different language groups across distinct linguistic dimensions. Our approach improves performance substantially over baselines. However, we find that linguistic relatedness—or even a lack thereof—does not covary meaningfully with adapter performance. Surprisingly, our cross-attention fine-tuning approach appears equally effective with randomly initialized adapters, implying that the benefit of adapters in this setting lies in parameter regularization, and not in meaningful information transfer. We provide analysis supporting this regularization hypothesis. Our findings underscore the reality that neural language processing involves many success factors, and that not all neural methods leverage linguistic knowledge in intuitive ways.

1 Introduction and Background

For a majority of the world's languages, data scarcity remains a barrier for reliable machine translation (MT). Cross-lingual transfer from high-resource to low-resource languages has been shown to help mitigate this (Zoph et al., 2016; Tars et al., 2021), but has still not closed the gap between high- and low-resource scenarios. In this work, we seek to exploit methods from parameter-efficient fine-tuning (PEFT) to further improve cross-lingual transfer for a Creoles, an under-served group of low-resource languages.

PEFT methods such as bottleneck adapters (Houlsby et al., 2019; Pfeiffer et al., 2020), LoRA

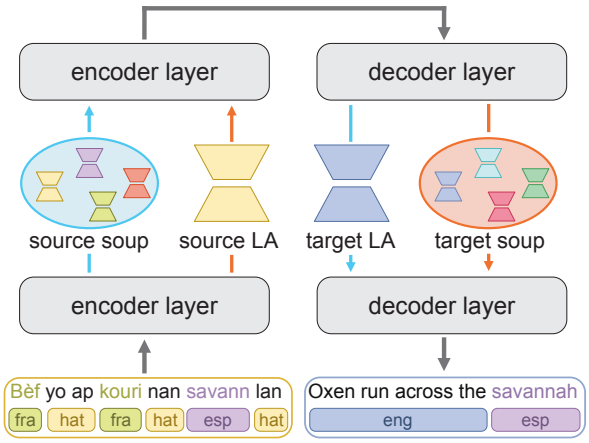


Figure 1: Overview of the MT transfer experiments between English and Creoles. The ↑ path shows the use of source language adapter in the encoder and the 'souped' (weight-averaged) target and transfer language adapters in the decoder layers. The ↑ path shows the opposite direction.

(Hu et al., 2021), and (IA)^3 (Liu et al., 2022a) are widely used to adapt language models to low-resource languages (Parović et al., 2022; Parovic et al., 2023; Rathore et al., 2023; Vázquez et al., 2024; Dementieva et al., 2025). This is in part because they are robust to hyperparameter choices and training data amounts, in contrast to full model fine-tuning (Pfeiffer et al., 2024). Some PEFT methods can successfully use monolingual data to improve MT performance; for example, Üstün et al. (2021) train denoising adapters for both source and target languages, inserting the adapters between the encoder and decoder layers of an MT model, respectively. By freezing all model weights except for the cross-attention of the decoder, they then carry out cross-attention fine-tuning (CA-FT) with only a limited amount of parallel data. Chronopoulou et al. (2023a) also address data scarcity in adaptation to new domains following Wortsman et al. (2022) by "souping" (*i.e.,* weight-space averaging) pretrained domain adapters, as

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 222–237
July 27 - August 1, 2025

in Equation 1 (where θ_i represent the weights of each domain adapter). We combine the principles of these prior works in our paper.

$$\theta_{soup} = \frac{1}{l}\sum_{i=1}^{l}\theta_i \quad (1)$$

Prior work has also catalyzed transfer by training language family adapters to leverage phylogenetic relations between languages (Faisal and Anastasopoulos, 2022; Chronopoulou et al., 2023b). Other factors of selecting transfer languages have been investigated in the past besides phylogeny. These include similarity in terms of explicit grammatical features like word order (Pires et al., 2019), similarity in terms of language representation space (Üstün et al., 2022), as well as lexical overlap (Pfeiffer et al., 2021; de Vries et al., 2022). Other metrics have been proposed to select candidate transfer languages, including Subword Evenness (SuE; Pelloni et al., 2022) which quantifies how uneven subword lengths are in a multilingual tokenizer: uneven implies successful transfer.

In this work we focus on MT for Creole languages, a set of low-resource languages with hundreds of millions of speakers who presently lack reliable MT systems, despite demonstrable need for them (Neto et al., 2020; Robinson et al., 2024). Creole language technologies have not been thoroughly researched or developed; yet several works that do exist inspired our decision to explore them (cf. our literature review in Appendix A). Although most Creoles are genealogically related to high-resource languages (*e.g.*, English, French, Spanish), previous works have had mixed success leveraging transfer learning to boost performance for Creoles (Lent et al., 2022a, 2024; Robinson et al., 2022, 2023). No prior work has conclusively determined which languages to select for successful Creole transfer, explored PEFT in this application, nor compared linguistic and non-linguistic language relations in Creole transfer language selection. These aspects combined make Creoles a viable testbed for comparing transfer criteria.

We apply a new adapter method inspired by Üstün et al.'s (2021) approach to MT for three Creole languages. (See details in §2.) We experimented with this approach to compare a large variety of dimensions of linguistic relatedness and found that, contrary to our initial hypotheses, gains are virtually indistinguishable regardless of which transfer criterion we apply. Still more surprisingly, incorporating unrelated languages or noise to the Creole adapters yields comparable improvements. We conjecture that even untrained adaptors may assist cross-attention fine-tuning by acting as regularizers. We couple this hypothesis with analysis presented in §3.

We contribute both ① a novel regularization method combining adapter souping with cross-attention fine-tuning that surpasses baselines by 8 BLEU in some settings, and ② evidence that some of the benefit of adapters lies in regularization, independent of a wide range of linguistic information.

2 Methodology and Experiments

We follow Üstün et al. (2021) using their two-step methodology. First, we train monolingual denoising adapters separately for each language. At inference time, we add the relevant adapters to the encoder and decoder corresponding to the source and target languages, and fine-tune the decoder cross-attention on small amounts of parallel bitexts. Furthermore, we induce transfer by weight-averaging ("souping") Creole adapters with candidate transfer language adapters inspired by previous efforts (Faisal and Anastasopoulos, 2024; Wortsman et al., 2022; Chronopoulou et al., 2023a). The novelty lies in combining source and target language adapters, cross-attention fine-tuning and souping in a single method for improving low-resource MT. (See more details about our methodology in Section 2.3).

We focus our efforts on three Creole languages: Haitian, Papiamento, and Sango, spoken in total by roughly 18 million speakers in Haiti, the ABC Islands, and the Central African Republic, respectively.[1] This selection is primarily due to the intersection of data availability and support by NLLB-200, the MT model we apply in experiments (NLLB Team et al., 2022). Despite their unique contexts, these languages share commonalities in socio-historical development: they evolved from linguistic contact between European and African languages resulting from Western colonialism (Kouwenberg and Singler, 2009). As such, they can trace parts of their lexicon and grammar back to both European and African languages.

2.1 Selecting Languages for Transfer

We examine five principled approaches to select transfer languages, which range from linguistically motivated to purely data-driven. First, we

[1] According to Ethnologue.

Dataset	Domain	Source
MADLAD	Web scrape	Kudugunta et al., 2023
NLLB-OPUS	Web scrape	NLLB Team et al., 2022
FLORES-200	Wikipedia	NLLB Team et al., 2022

Table 1: Datasets and domains used. We use MADLAD to train adapters, NLLB-OPUS to fine-tune cross-attention, and FLORES-200 to evaluate.

leverage language phylogeny, selecting likely Indo-European (**IE**; Rickford and McWhorter, 1997; Lefebvre, 2011) and Niger-Congo (**NC**; Hammarström et al., 2023; Mufwene, 2006; Hall, 1958; Faine, 1937) **relatives** for Haitian, Papiamento, and Sango. We also experiment with transfer between these **Creole languages**, as previous works have argued that Creole languages have linguistic commonalities with each other (Daval-Markussen and Bakker, 2012). We next select languages by similarity of documented typological features, calculated as `lang2vec` distances (Littell et al., 2017). For data-driven language selection, we determine language relatedness from similarity of NLLB-200 language embeddings (**NLLB representations**; NLLB Team et al., 2022), as well as Subword Evenness (**SuE**; Pelloni et al., 2022). For details of our language selection processes, see Appendix C.

2.2 Control experiments

As an experimental control, we identify three language groups with no linguistic relation to the Creole languages in question: **Uralic** languages, **Dravidian** languages, and **CJK** (*i.e.*, Chinese, Japanese, and Korean). None of these groups share phylogeny or significant amounts of typological features with our Creole languages. The latter two also differ in terms of script. And the final group is distinct in that its languages also bear no phylogenetic relation to each other. As a final experimental control, we use a randomly initialized, untrained adapter (`init`). We use this untrained adapter in two ways: as a replacement for the Creole language adapter, and souped with the Creole language adapter. As Chronopoulou et al. (2023a) suggest for souping, we use the same random initialization for all untrained and trained adapters.

2.3 Experimental Details

Our experiments follow the same general steps. ① We train all language adapters individually on denoising using monolingual data from MADLAD-400 (Kudugunta et al., 2023) that we preprocess and clean further.[2] ② We add the source language adapter s to the encoder and the target language adapter t to the decoder and freeze all adapter and model parameters. In adapter averaging experiments the `init` or the helper language adapters are souped with the s or t language adapters in this step as well. ③ We unfreeze the cross-attention of the decoder and use 10k parallel data from NLLB-OPUS (NLLB Team et al., 2022) to do cross-attention fine-tuning (CA-FT). ④ We evaluate MT from $s \rightarrow t$ on Flores-200 (NLLB Team et al., 2022) using mean BLEU and chrF scores. (See Table 1 for data composition.)

All experiments are conducted using a 600M-parameter distillation of the original 54B-parameter NLLB-200 model (NLLB Team et al., 2022) which has 12 encoder and 12 decoder layers, 16 attention heads, 1,024 dimensions, and a model vocabulary of over 250,000 sentence-piece tokens shared across all languages. We adopt the adapter training and the hyperparameters from Üstün et al. (2021), training bottleneck adapters on 10,000 training samples for a maximum of 100,000 training steps with early stopping, a batch size of 8, 4,000 warmup steps, and the Adam optimizer with a maximum learning rate of $2e-4$ with a gradient accumulation of 8 steps. All adapters initialized from the same random seed as Chronopoulou et al. (2023a) show that this is necessary for successful souping. We train CA-FT for 1 epoch with early stopping. We soup adapters with the same weight except when averaging with the `init` adapter where the weight of the Creole language to `init` adapter is $1:3$, to closer approximate the other setups where the Creole language adapter is souped with three other adapters.

3 Results and Analysis

Here we present only Creole-to-English BLEU scores. We conducted preliminary experiments using only 800 aligned fine-tuning segments on these and the inverse translation directions, and saw the same trends shown in our results here. We decided to prioritize into-English MT for our main experiments after seeing small adapter gains across the board for into-Creole. (See Appendix D for our full results suite.) Table 2 shows BLEU improvements of our approaches over both baselines: the base MT model with CA-FT, and the s and t language adapters without souping (Üstün et al., 2021). The

[2] See Appendix B for preprocessing details.

Experiment	hat → eng	pap → eng	sag → eng
Base Model (CA-FT)	33.37	38.97	10.89
s and *t* Adapters	32.33	40.04	11.40
📻 Untrained *s* Adapter	37.07	45.01	14.91
🌳 IE Transfer	36.44	46.35	12.46
🌳 NC Transfer	36.06	46.69	12.29
❤️ Creole Transfer	35.25	46.23	12.76
📖 lang2vec	36.54	47.04	13.07
👗 NLLB Vec	35.80	46.91	12.80
⚖️ Subword Evenness	36.36	47.03	13.12
Mean Improvement	+2.70	+7.74	+1.86
📻 Untrained Souping	37.42	46.34	13.41
🖼️ Uralic	37.06	47.00	13.58
🖼️ CJK	36.41	47.17	13.33
🖼️ Dravidian	36.55	47.27	13.27
Mean Improvement	+3.49	+7.98	+2.51

Table 2: Mean BLEU scores for Creole to English MT experiments over a diverse set of conditions for selecting transfer languages, as motivated by phylogeny (🌳), language group (❤️), typological databases (📖), model representations (👗), and tokenization (⚖️). Experimental results are also compared against control experiments with unrelated languages (🖼️) and untrained adapters (📻). We highlight mean performance improvements over the Base Model (CA-FT) for both transfer and controls.

improvements, however, are not particularly differentiated between various transfer and control scenarios, and notably, untrained adapters match or exceed performance of trained ones.

Regularization, Not Transfer? Two facets of our results indicate that cross-lingual transfer does not explain the success of our methodology. The first is that adapter souping improves performance over the baselines, *regardless* of the criteria used to select transfer languages. The unrelated control languages (*i.e.*, Dravidian, Uralic, and CJK) achieve performance gains on par with languages selected through principled means (*e.g.*, genealogy and typology), without marked differences in BLEU or chrF scores. This holds for both the source and target language adapters in the encoder and decoder, respectively. The second is the commensurate success of the untrained adapters in averaging. Souping adapters with 'meaningful' language adapters is no better than souping with random noise, indicating that cross-lingual transfer (or any information transfer) is likely not responsible for performance gains. These findings seem more consistent with previous works on regularization, which find that adding noise can improve a model's ability to generalize and avoid overfitting (Hochreiter and Schmidhuber, 1994; Poole et al., 2014; Moradi et al., 2020) (which may be pertinent given both our limited train sets and train/test domain mismatch).

FT examples:	800	10k
Base Model (CA-FT)	45.45	45.53
s and *t* Adapters	38.58	41.92
🌳 `spa + por + oci` Souping	41.87	43.74
📻 Untrained Souping	43.97	44.75

Table 3: BLEU scores for Catalan to English MT.

We perform multiple analyses to test our hypothesis that adapters are regularizing. First, we compared the use of untrained adapters to the use of adapters of known closely related languages. Because the close relatives of Creole languages are disputed, we turned to another relatively low resource language, Catalan, which has three close relatives also supported by NLLB-200. Spanish, Portuguese, and Occitan are closely related to Catalan by effectively every reasonable linguistic criterion. Table 3 shows different trends for Catalan (a language on which NLLB's baseline performance is notably high). In this scenario, none of the adapter approaches improve performance over the baseline, once again indicating that the monolingual denoising adapter training on MADLAD data is not positively contributing. And once again, even with these highly attested close linguistic relationships, we see the random adapter comes nearest to recovering the baseline score in both data settings. This set of results underscores the possibility that CA-FT without randomized regularization may lead to overfitting (hence the performance drops). We also experimented souping only one or two related languages for both Catalan→English and Papiamento→English but saw no improvements.

We observed training loss, gradient norms, and dev loss for CA-FT of the experiments presented in this section. We compare untrained adapter souping (hypothesized regularization) with simple *s* and *t* adapters (unregularized), finding that the unregularized case has moderately higher gradient norms and dev loss, consistent with our hypothesis. The unregularized case also sees higher train loss, which is inconclusive regarding our hypothesis since an overfit model may be stuck in a local minimum or be overfit on a distribution varying from the CA-FT data. (See Appendix E.)

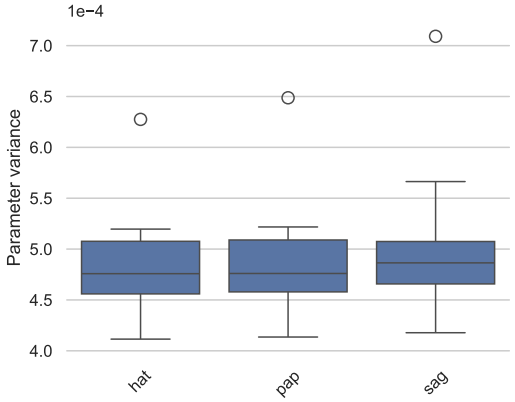


Figure 2: Parameter variance between the pretrained Creole adapters (represented by the outliers) vs the souped adapters of various experimental conditions (shown in the boxplots).

We also show that parameter variance decreases substantially when souping adapters compared to the pretrained Creole adapters (Figure 2), further supporting our regularization explanation.

Qualitative Evaluation Beyond computational analyses, we further assess the quality of translations across our experimental conditions. To start, a subset of the authors create a simple annotation guideline, inspired by MQM (Burchardt, 2013), to help identify errors regarding vocabulary, grammar, and other translation issues (*e.g.*, addition or omission). Then, one author, a native Haitian speaker, carries out a linguistically-grounded manual evaluation of Haitian → English translation across the baseline of s and t Adapters, the IE Transfer, and the Untrained Souping. The evaluation involves 33 samples per condition presented to the annotator in three shuffled sets (*i.e.*, 99 sentences in total) to avoid biasing him. The primarily focus falls on grammar and translation issues (For a condensed list of errors, see Table 9 in Appendix F). Although a small sample size, we observe that adapter souping results in less grammatical and lexical errors when applying the pretrained Haitian and English adapters with CA-FT. While both IE Transfer and Untrained Souping are comparable in reducing lexical errors, the latter does better in avoiding grammatical errors. This may indicate a comparative success of regularization when adding random noise that is also reflected in the $+0.98$ BLEU score Untrained Souping has over IE Transfer for Haitian → English (see Table 2). We believe that even a small manual translation evaluation demonstrates the value of involving native speakers in MT.

4 Conclusion

Inspired by prior success of monolingual adapters and CA-FT for low-resource MT, we corroborate prior findings that adapters can help by consistently improving BLEU scores for a set of Creole languages. However, we differ from prior research by finding no evidence of linguistic or informational transfer from adapters, even across a wide array of linguistic relationships. We hypothesize that most adapter aid in this setting is from regularization and find some analytical support for this.

Limitations

Scalability This work can only be applied to a small subset of all Creoles that are present in larger benchmarks (Lent et al., 2024; Robinson et al., 2024) due to the fact that they are low-resource languages – in certain cases, extremely so –, so there is an issue of data scarcity and a limited support by MT models. This data scarcity extends to also the African relatives of the Creoles we target in this paper. Our work is further limited by the limited coverage by typological resources. Grambank (Skirgård et al., 2023) supports only Haitian, not Papiamento and Sango, and while the lang2vec covers more languages, the coverage is limited and uneven. Despite these constraints, however, we believe our selection of Creoles (*i.e.*, Haitian, Papiamento, and Sango) are diverse enough to make high-level hypotheses about transfer learning for Creoles; these languages have different relatives (*e.g.*, inheriting large vocabulary from French, Portuguese, and Ngbandi, respectively) and are spoken across different regions (*i.e.*, the Caribbean and Africa). Moreover, our experimentation over these languages is **extensive**, with careful investigation of 6 transfer scenarios, 1 `init` baseline, and 3 principled controls. So while this work cannot scale *widely* across Creoles, we investigate transfer scenarios for each Creole *deeply*.

Domain of Data Similar to many other low-resource languages, much of the available data for Creoles comes from the religious domain (*e.g.*, Bible translations), which has a characteristic style that is typically not representative of language use in other contexts (Mielke et al., 2019). Moreover, religious domain content may not necessarily be culturally appropriate for some communities (Hershcovich et al., 2022; Mager et al., 2023). In order to be truly useful for end users, data sourced from a

wide variety of domains will ultimately be a necessary component for the development of robust MT systems for Creoles. Until more such data becomes available, we believe that work at the intersection of transfer learning and domain adaptation could be greatly beneficial for Creole NLP.

Difficulty in Verifying Neural Phenomena While we find some analytical evidence that our adapters are boosting performance by regularizing fine-tuning, it is difficult to verify this with certainty. Regularization is a mathematical phenomenon that can be demonstrated rigorously, but such involved analysis is outside the scope of this current work; hence we leave these important analytical details to future work.

Ethics Statement

As Creole languages are often marginalized, we recognize the necessity of heightened ethical scrutiny for research involving these languages. We thus engage directly with considerations from Lent et al. (2022b), Mager et al. (2023), and Bird (2020), to ensure that our work is in accordance with the needs and desires of speakers. To this end, we note that speakers of many different Creole languages actively *want* and/or *need* improved MT technology for their language. In the context of Haitian Creole, for example, MT technology is highly desirable for facilitating interactions with non-Haitian speakers (*e.g.*, the neighboring, Spanish-speaking Dominican Republic, and foreign workers at NGOs operating in Haiti). Additionally, MT has the potential to greatly improve access to information for Haitians. DeGraff et al. (2022) argues that the lack of access to education in Creole is a critical barrier to social and economic mobility in Haiti. Accordingly, reliable MT for Haitian Creole could make available high-quality educational resources from platforms such as edX, Coursera, MITx, and Udemy, for example. Thus, if MT systems could offer accurate translations into Haitian Creole, they would become transformative tools for advancing the country's development, breaking down educational barriers, and fostering greater social equity. We note similar potential for MT for speakers of Sango in Central African Republic, Chad, and Democratic Republic of the Congo and (African Development Bank Group, 2024). And while the socioeconomic context of Aruba, Bonaire, and Curacao is quite different from that of aforementioned Haiti or Central African Republic, speakers' desire for language technology for Papiamento is well documented through projects like papiamentu.ai.

Acknowledgments

MF, JB, and HL were supported by the Carlsberg Foundation, under the *Semper Ardens: Accelerate* program (project nr. CF210454); the same project also supported the annotation work performed by DJB. EL was funded by the Google Award for Inclusion Research program (awarded to JB and HL for the "CREOLE: Creating Resources for Disadvantaged Language Communities" project). We extend our thanks to Mike Zhang, Esther Ploeger, and the other members of AAU-NLP, for their helpful feedback on our manuscript.

References

David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Muhammad, Guyo Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme, Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Gitau, Jade Abbott, Mohamed Ahmed, Millicent Ochieng, Anuoluwapo Aremu, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, Andiswa Bukula, and Sam Manthalu. 2022. A few thousand translations go a long way! leveraging pre-trained models for African news translation. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3053–3070, Seattle, United States. Association for Computational Linguistics.

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen H. Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, Seid Muhie Yimam, Tajuddeen Rabiu Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verrah Otiende, Iroro Orife, Davis David, Samba Ngom, Tosin Adewumi, Paul Rayson, Mofetoluwa Adeyemi, Gerald Muriuki, Emmanuel Anebi, Chiamaka Chukwuneke, Nkiruka Odu, Eric Peter Wairagala, Samuel Oyerinde, Clemencia Siro, Tobius Saul Bateesa, Temilola Oloyede, Yvonne Wambui, Victor Akin-

ode, Deborah Nabagereka, Maurice Katusiime, Ayodele Awokoya, Mouhamadane MBOUP, Dibora Gebreyohannes, Henok Tilaye, Kelechi Nwaike, Degaga Wolde, Abdoulaye Faye, Blessing Sibanda, Orevaoghene Ahia, Bonaventure F. P. Dossou, Kelechi Ogueji, Thierno Ibrahima DIOP, Abdoulaye Diallo, Adewale Akinfaderin, Tendai Marengereke, and Salomey Osei. 2021. MasakhaNER: Named entity recognition for African languages. Transactions of the Association for Computational Linguistics, 9:1116–1131.

African Development Bank Group. 2024. Central african republic: New digital era dawns with debut of high-speed internet cable. African Development Bank Group.

Hamidreza Amirzadeh, Sadegh Jafari, Anika Harju, and Rob van der Goot. 2025. data2lang2vec: Data driven typological features completion. In Proceedings of the 31st International Conference on Computational Linguistics, pages 6520–6529, Abu Dhabi, UAE. Association for Computational Linguistics.

Ruth-Ann Armstrong, John Hewitt, and Christopher Manning. 2022. JamPatoisNLI: A jamaican patois natural language inference dataset. In Findings of the Association for Computational Linguistics: EMNLP 2022, pages 5307–5320, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Steven Bird. 2020. Decolonising speech and language technology. In Proceedings of the 28th International Conference on Computational Linguistics, pages 3504–3519, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Aljoscha Burchardt. 2013. Multidimensional quality metrics: a flexible system for assessing translation quality. In Proceedings of Translating and the Computer 35, London, UK. Aslib.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In Proceedings of the Sixth Workshop on Statistical Machine Translation, pages 22–64, Edinburgh, Scotland. Association for Computational Linguistics.

Bernard Caron, Marine Courtin, Kim Gerdes, and Sylvain Kahane. 2019. A surface-syntactic UD treebank for Naija. In Proceedings of the 18th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019), pages 13–24, Paris, France. Association for Computational Linguistics.

Alexandra Chronopoulou, Matthew Peters, Alexander Fraser, and Jesse Dodge. 2023a. AdapterSoup: Weight averaging to improve generalization of pretrained language models. In Findings of the Association for Computational Linguistics: EACL 2023, pages 2054–2063, Dubrovnik, Croatia. Association for Computational Linguistics.

Alexandra Chronopoulou, Dario Stojanovski, and Alexander Fraser. 2023b. Language-family adapters for low-resource multilingual neural machine translation. In Proceedings of the Sixth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2023), pages 59–72, Dubrovnik, Croatia. Association for Computational Linguistics.

Christopher Clarke, Roland Daynauth, Jason Mars, Charlene Wilkinson, and Hubert Devonish. 2024. GuyLingo: The Republic of Guyana creole corpora. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 792–798, Mexico City, Mexico. Association for Computational Linguistics.

Raj Dabre and Aneerav Sukhoo. 2022. KreolMorisienMT: A dataset for mauritian creole machine translation. In Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022, pages 22–29, Online only. Association for Computational Linguistics.

Aymeric Daval-Markussen and Peter Bakker. 2012. Explorations in creole research with phylogenetic tools. In Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH, pages 89–97, Avignon, France. Association for Computational Linguistics.

Wietse de Vries, Martijn Wieling, and Malvina Nissim. 2022. Make the best of cross-lingual transfer: Evidence from POS tagging with over 100 languages. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7676–7685, Dublin, Ireland. Association for Computational Linguistics.

Michel DeGraff, William Scott Frager, and Haynes Miller. 2022. Language policy in haitian education. Journal of Haitian Studies, 28(2):33–95.

Daryna Dementieva, Valeriia Khylenko, and Georg Groh. 2025. Cross-lingual text classification transfer: The case of Ukrainian. In Proceedings of the 31st International Conference on Computational Linguistics, pages 1451–1464, Abu Dhabi, UAE. Association for Computational Linguistics.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. WALS Online (v2020.3). Zenodo.

Jules Faine. 1937. Philologie creole: Études historique et étymologiques sur la langue creoles d'Haïti. Imprimerie de l'Etat, Port-au-Prince.

Fahim Faisal and Antonios Anastasopoulos. 2022. Phylogeny-inspired adaptation of multilingual models to new languages. In Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 434–452, Online only. Association for Computational Linguistics.

Fahim Faisal and Antonios Anastasopoulos. 2024. Data-augmentation-based dialectal adaptation for LLMs. In Proceedings of the Eleventh Workshop on NLP for Similar Languages, Varieties, and Dialects (VarDial 2024), pages 197–208, Mexico City, Mexico. Association for Computational Linguistics.

Robert E. Frederking, Ralf D. Brown, and Christopher Hogan. 1997. The DIPLOMAT rapid development speech MT system. In Proceedings of Machine Translation Summit VI: Systems, pages 261–262, San Diego, California.

Robert Hall. 1958. Creole languages and genetic relationships. Word, 14:367–373.

Harald Hammarström, Sebastian Nordhoff, Robert Forkel, Martin Haspelmath, and Sebastian Bank. 2023. Glottolog. Online database.

Daniel Hershcovich, Stella Frank, Heather Lent, Miryam de Lhoneux, Mostafa Abdou, Stephanie Brandl, Emanuele Bugliarello, Laura Cabello Piqueras, Ilias Chalkidis, Ruixiang Cui, Constanza Fierro, Katerina Margatina, Phillip Rust, and Anders Søgaard. 2022. Challenges and strategies in cross-cultural NLP. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6997–7013, Dublin, Ireland. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1994. Simplifying neural nets by discovering flat minima. In Advances in Neural Information Processing Systems, volume 7. MIT Press.

Neil Houlsby, Andrei Giurgiu, Stanisław Jastrzebski, and Bruna Morrone. 2019. Parameter-Efficient Transfer Learning for NLP. In Proceedings of the 36th International Conference on Machine Learning, page 10, Long Beach, California.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv preprint. ArXiv:2106.09685 [cs].

Silvia Kouwenberg and John Victor Singler. 2009. The Handbook of Pidgin and Creole Studies. John Wiley & Sons.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Christopher A. Choquette-Choo, Katherine Lee, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2023. Madlad-400: A multilingual and document-level large audited dataset. Preprint, arXiv:2309.04662.

Claire Lefebvre. 2011. Substrate features in the properties of verbs in three atlantic creoles: Haitian Creole, Saramaccan and Papiamentu. In Claire Lefebvre, editor, Creoles, their Substrates, and Language Typology, pages 127–154. John Benjamins, Amsterdam.

Heather Lent, Emanuele Bugliarello, and Anders Søgaard. 2022a. Ancestor-to-creole transfer is not a walk in the park. In Proceedings of the Third Workshop on Insights from Negative Results in NLP, pages 68–74, Dublin, Ireland. Association for Computational Linguistics.

Heather Lent, Kelechi Ogueji, Miryam de Lhoneux, Orevaoghene Ahia, and Anders Søgaard. 2022b. What a creole wants, what a creole needs. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 6439–6449, Marseille, France. European Language Resources Association.

Heather Lent, Kushal Tatariya, Raj Dabre, Yiyi Chen, Marcell Fekete, Esther Ploeger, Li Zhou, Ruth-Ann Armstrong, Abee Eijansantos, Catriona Malau, Hans Erik Heje, Ernests Lavrinovics, Diptesh Kanojia, Paul Belony, Marcel Bollmann, Loïc Grobol, Miryam de Lhoneux, Daniel Hershcovich, Michel DeGraff, Anders Søgaard, and Johannes Bjerva. 2024. CreoleVal: Multilingual multitask benchmarks for creoles. Transactions of the Association for Computational Linguistics, 12:950–978.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022a. Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning. arXiv preprint. ArXiv:2205.05638 [cs].

Zhengyuan Liu, Shikang Ni, Ai Ti Aw, and Nancy F. Chen. 2022b. Singlish message paraphrasing: A joint task of creole translation and text normalization. In Proceedings of the 29th International Conference on Computational Linguistics, pages 3924–3936, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Manuel Mager, Elisabeth Mager, Katharina Kann, and Ngoc Thang Vu. 2023. Ethical considerations for machine translation of indigenous languages: Giving a voice to the speakers. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4871–4897, Toronto, Canada. Association for Computational Linguistics.

Susanne Maria Michaelis, Philippe Maurer, Martin Haspelmath, and Magnus Huber, editors. 2013. APiCS Online. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Sabrina J. Mielke, Ryan Cotterell, Kyle Gorman, Brian Roark, and Jason Eisner. 2019. What kind of language is hard to language-model? In Proceedings

of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4975–4989, Florence, Italy. Association for Computational Linguistics.

Ludovic Mompelat, Daniel Dakota, and Sandra Kübler. 2022. How to parse a creole: When martinican creole meets French. In Proceedings of the 29th International Conference on Computational Linguistics, pages 4397–4406, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Reza Moradi, Reza Berangi, and Behrouz Minaei. 2020. A survey of regularization strategies for deep models. Artificial Intelligence Review, 53(6):3947–3986.

Salikoko Mufwene. 2006. Pidgins and creoles. In Braj B. Kachru, Yamuna Kachru, and Cecil L. Nelson, editors, The Handbook of World Englishes, pages 313–327. Blackwell, Hoboken, NJ.

Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Seid Muhie Yimam, David Ifeoluwa Adelani, Ibrahim Said Ahmad, Nedjma Ousidhoum, Abinew Ali Ayele, Saif Mohammad, Meriem Beloucif, and Sebastian Ruder. 2023. SemEval-2023 task 12: Sentiment analysis for African languages (AfriSenti-SemEval). In Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 2319–2337, Toronto, Canada. Association for Computational Linguistics.

Shamsuddeen Hassan Muhammad, David Ifeoluwa Adelani, Sebastian Ruder, Ibrahim Sa'id Ahmad, Idris Abdulmumin, Bello Shehu Bello, Monojit Choudhury, Chris Chinenye Emezue, Saheed Salahudeen Abdullahi, Anuoluwapo Aremu, Alípio Jorge, and Pavel Brazdil. 2022. NaijaSenti: A Nigerian Twitter sentiment corpus for multilingual sentiment analysis. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 590–602, Marseille, France. European Language Resources Association.

Antonio Carvalho Neto, Fernanda Versiani, Kelly Pellizari, Carolina Mota-Santos, and Gustavo Abreu. 2020. Latin american, african and asian immigrants working in brazilian organizations: facing the language barrier. Revista Economia & Gestão, 20(55).

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia-Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Kelechi Ogueji and Orevaoghene Ahia. 2019. Pidginunmt: Unsupervised neural machine translation from west african pidgin to english. arXiv preprint arXiv:1912.03444.

Wuraola Fisayo Oyewusi, Olubayo Adekanmbi, and Olalekan Akinsande. 2020. Semantic enrichment of nigerian pidgin english for contextual sentiment classification. arXiv preprint arXiv:2003.12450.

Marinela Parovic, Alan Ansell, Ivan Vulić, and Anna Korhonen. 2023. Cross-lingual transfer with target language-ready task adapters. In Findings of the Association for Computational Linguistics: ACL 2023, pages 176–193, Toronto, Canada. Association for Computational Linguistics.

Marinela Parović, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2022. BAD-X: Bilingual adapters improve zero-shot cross-lingual transfer. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1791–1799, Seattle, United States. Association for Computational Linguistics.

Olga Pelloni, Anastassia Shaitarova, and Tanja Samardzic. 2022. Subword evenness (SuE) as a predictor of cross-lingual transfer to low-resource languages. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7428–7445, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Maria Ponti. 2024. Modular deep learning. Preprint, arXiv:2302.11529.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 7654–7673, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Edoardo Maria Ponti, Helen O'Horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, Thierry Poibeau, Ekaterina Shutova, and Anna Korhonen. 2019. Modeling language variation and universals: A survey on typological linguistics for natural language processing. Computational Linguistics, 45(3):559–601.

Ben Poole, Jascha Sohl-Dickstein, and Surya Ganguli. 2014. Analyzing noise in autoencoders and deep networks. CoRR, abs/1406.1831.

Vipul Rathore, Rajdeep Dhingra, Parag Singla, and Mausam. 2023. ZGUL: Zero-shot generalization to unseen languages using multi-source ensembling of language adapters. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 6969–6987, Singapore. Association for Computational Linguistics.

John R Rickford and John McWhorter. 1997. Language contact and language generation: Pidgins and creoles. In Florian Coulmas, editor, The handbook of sociolinguistics, pages 238–256. Blackwell, Malden, MA.

Nathaniel Robinson, Raj Dabre, Ammon Shurtz, Rasul Dent, Onenamiyi Onesi, Claire Monroc, Loïc Grobol, Hasan Muhammad, Ashi Garg, Naome Etori, Vijay Murari Tiyyala, Olanrewaju Samuel, Matthew Stutzman, Bismarck Odoom, Sanjeev Khudanpur, Stephen Richardson, and Kenton Murray. 2024. Kreyòl-MT: Building MT for Latin American, Caribbean and colonial African creole languages. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 3083–3110, Mexico City, Mexico. Association for Computational Linguistics.

Nathaniel Robinson, Cameron Hogan, Nancy Fulda, and David R. Mortensen. 2022. Data-adaptive transfer learning for translation: A case study in Haitian and jamaican. In Proceedings of the Fifth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2022), pages 35–42, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Nathaniel Romney Robinson, Matthew Dean Stutzman, Stephen D. Richardson, and David R Mortensen. 2023. African substrates rather than european lexifiers to augment african-diaspora creole translation. In 4th Workshop on African Natural Language Processing.

Hedvig Skirgård, Hannah J. Haynie, Damián E. Blasi, Harald Hammarström, Jeremy Collins, Jay J. Latarche, Jakob Lesage, Tobias Weber, Alena Witzlack-Makarevich, Sam Passmore, Angela Chira, Luke Maurits, Russell Dinnage, Michael Dunn, Ger Reesink, Ruth Singer, Claire Bowern, Patience Epps, Jane Hill, Outi Vesakoski, Martine Robbeets, Noor Karolin Abbas, Daniel Auer, Nancy A. Bakker, Giulia Barbos, Robert D. Borges, Swintha Danielsen, Luise Dorenbusch, Ella Dorn, John Elliott, Giada Falcone, Jana Fischer, Yustinus Ghanggo Ate, Hannah Gibson, Hans-Philipp Göbel, Jemima A. Goodall, Victoria Gruner, Andrew Harvey, Rebekah Hayes, Leonard Heer, Roberto E. Herrera Miranda, Nataliia Hübler, Biu Huntington-Rainey, Jessica K. Ivani, Marilen Johns, Erika Just, Eri Kashima, Carolina Kipf, Janina V. Klingenberg, Nikita König, Aikaterina Koti, Richard G. A. Kowalik, Olga Krasnoukhova, Nora L.M. Lindvall, Mandy Lorenzen, Hannah Lutzenberger, Tônia R.A. Martins, Celia Mata German, Suzanne van der Meer, Jaime Montoya Samamé, Michael Müller, Saliha Muradoğlu, Kelsey Neely, Johanna Nickel, Miina Norvik, Cheryl Akinyi Oluoch, Jesse Peacock, India O.C. Pearey, Naomi Peck, Stephanie Petit, Sören Pieper, Mariana Poblete, Daniel Prestipino, Linda Raabe, Amna Raja, Janis Reimringer, Sydney C. Rey, Julia Rizaew, Eloisa Ruppert, Kim K. Salmon, Jill Sammet, Rhiannon Schembri, Lars Schlabbach, Frederick W.P. Schmidt, Amalia Skilton, Wikaliler Daniel Smith, Hilário de Sousa, Kristin Sverredal, Daniel Valle, Javier Vera, Judith Voß, Tim Witte, Henry Wu, Stephanie Yam, Jingting Ye, Maisie Yong, Tessa Yuditha, Roberto Zariquiey, Robert Forkel, Nicholas Evans, Stephen C. Levinson, Martin Haspelmath, Simon J. Greenhill, Quentin D. Atkinson, and Russell D. Gray. 2023. Grambank reveals global patterns in the structural diversity of the world's languages. Science Advances, 9.

Maali Tars, Andre Tättar, and Mark Fišel. 2021. Extremely low-resource machine translation for closely related languages. In Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa), pages 41–52, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Ahmet Üstün, Alexandre Berard, Laurent Besacier, and Matthias Gallé. 2021. Multilingual unsupervised neural machine translation with denoising adapters. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6650–6662, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, Gertjan van Noord, and Sebastian Ruder. 2022. Hyper-X: A unified hypernetwork for multi-task multilingual transfer. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7934–7949, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Raúl Vázquez, Timothee Mickus, Jörg Tiedemann, Ivan Vulić, and Ahmet Üstün, editors. 2024. Proceedings of the 1st Workshop on Modular and Open Multilingual NLP (MOOMIN 2024). Association for Computational Linguistics, St Julians, Malta.

Hongmin Wang, Yue Zhang, GuangYong Leonard Chan, Jie Yang, and Hai Leong Chieu. 2017. Universal Dependencies parsing for colloquial Singaporean English. In Proceedings of the 55th Annual Meeting

of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1732–1744, Vancouver, Canada. Association for Computational Linguistics.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pages 23965–23998. PMLR.

Fei Yuan, Yinquan Lu, Wenhao Zhu, Lingpeng Kong, Lei Li, Yu Qiao, and Jingjing Xu. 2023. Lego-MT: Learning detachable models for massively multilingual machine translation. In Findings of the Association for Computational Linguistics: ACL 2023, pages 11518–11533, Toronto, Canada. Association for Computational Linguistics.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

Just Zwennicker and David Stap. 2022. Towards a general purpose machine translation system for sranantongo. arXiv preprint arXiv:2212.06383.

A Literature Review

NLP research for Creole languages is still in relatively early stages. Some large-scale multilingual MT systems have limited Creole language support: NLLB (NLLB Team et al., 2022) supports five Creole languages, and Google Translate[3] and Lego-MT (Yuan et al., 2023) each support eight. Other works have focused specifically on building MT for Creole languages Haitian (Robinson et al., 2022; Callison-Burch et al., 2011; Frederking et al., 1997), Singlish (Wang et al., 2017; Liu et al., 2022b), Nigerian Pidgin (Adelani et al., 2022; Ogueji and Ahia, 2019), Mauritian Creole (Dabre and Sukhoo, 2022), Sranan Tongo (Zwennicker and Stap, 2022), Guyanese Creole (Clarke et al., 2024), and Jamaican Patois (Robinson et al., 2022). Some others have focused on building Creole language-specific NLP tools outside of MT for Nigerian Pidgin (Muhammad et al., 2023, 2022; Adelani et al., 2021; Oyewusi et al., 2020; Caron et al., 2019), Jamaican Patois (Armstrong et al., 2022), and Martinican Creole (Mompelat et al., 2022).

More recently, researchers have developed large scale multi-Creole-language NLP applications. Lent et al. (2024) developed CreoleVal, a multitask benchmark for Creoles including an MT dataset and model for 28 Creole languages. Robinson et al. (2024) expanded on Lent et al.'s (2024) work, building genre-diverse MT datasets and models for 41 Creole languages of the Caribbean, Latin America, and Colonial Africa. Perhaps most pertinent to our current work are past studies into cross-lingual transfer to benefit Creole NLP. In tightly controlled experiments, Robinson et al. (2023) found indications that Haitian and Jamaican Patois' Niger-Congo relatives may be more helpful as transfer languages than their Indo-European relatives, for MT applications. Lent et al. (2022a), however, did not identify clear advantages from a language's relatedness to a Creole language in terms of its usefulness in transfer learning. Given the sparse research and even sparser positive results in the area, optimal transfer learning for Creole NLP remains an open question.

B Data Preprocessing

The MadLad corpus is organized in series of internet-scraped documents, which the MadLad team cross-checks for noise (Kudugunta et al., 2023). For training the monolingual adapters, we use the *MadLad-Clean* subsplit and we perform further preprocessing, since we noted after manual inspection that noise can still be present in the documents. Firstly, the MadLad documents are split into sentences using the SentenceSplitter [4] library.

For each sentence in the document, we define the following rules, if any of the rules is breached, then the given sentence is not included in the final monolingual training corpus.

1. Sentence ends with a terminal punctuation mark ('.'; '!', ' " ', '?').
2. Ratio of numerical characters with respect to all other characters is less than **0.25**.
3. Sentence starting with a bullet-point.
4. Sentence containing a *lorem ipsum* substring.
5. Ratio of stop-words with respect to all other tokens is less than **0.6** using whitespace tokenization.
6. Ratio of token count that does not contain alpha characters with respect to total token count is less than **0.25**, using whitespace tokenization.

[3] https://translate.google.com/ as of 8 Feb., 2025

[4] https://github.com/mediacloud/sentence-splitter

7. Sentence length is less than **14** and more than **300** characters.
8. Ratio of capitalized characters is less than **0.2**.
9. Ratio of alpha characters with respect to all other characters is more than **0.6**.
10. Language identification does not match the expected language. For this we employ FastText model [5].

This preprocessing removed approximately 20-23% of sentences, depending on the language. The same preprocessing steps were also applied to NLLB parallel data used for cross-attention fine-tuning, where preprocessing removed approximately 30-50% of the sentences.

C Selecting Languages for Transfer

Table 4 displays the different transfer languages we selected in our experiments. Below we detail our processes of selecting them.

C.1 Phylogeny-based Transfer

Faisal and Anastasopoulos (2022) followed principles of language phylogenetics to inform their transfer learning experiments, training adapters on combined Indo-European data, and for subfamilies of Indo-European such as Germanic, Romance, Indo-Aryan, etc. To translate for French, for example, they stacked a French adapter on top of a Romance and then an Indo-European adapter at inference time. Creole phylogeny is heavily disputed, however (Kouwenberg and Singler, 2009). Linguists have debated whether Atlantic Creole languages such as Haitian and Papiamento are best understood as Niger-Congo (NC) languages with extensive Indo-European (IE) lexical borrowings (Lefebvre, 2011; Rickford and McWhorter, 1997), or as IE languages that developed with strong NC influence (Faine, 1937; Hall, 1958; Hammarström et al., 2023; Mufwene, 2006). Considering this lack of consensus, we select two groups of related languages for each Creole language: IE relatives and NC relatives. In this work, we made a selection based on APICS (Michaelis et al., 2013) and linguistic references.

C.2 Typological Feature-based Transfer

Information about the typological features of languages is typically documented in databases such as APICS (Michaelis et al., 2013), WALS (Dryer

Condition	Language
Haitian	
Creole Language	hat
IE Relative	fr, es, en
NC Relative	ig, ee, fon
`lang2vec`	grc, oc, fo
NLLB	pap, epo, ltz
Papiamento	
Creole Language	pap
IE Relative	nl, pt, es
NC Relative	kg, ak, fon
`lang2vec`	suz, sr, ug
NLLB	hat, war, tgl
Sango	
Creole Language	sag
IE Relative	fr
NC Relative	kbp
`lang2vec`	ig, yo, ca
NLLB	kg, tsn, lin
Subword Evenness	kat, rus, zsm
Controls	
Uralic	fin, est, hun
CJK	zho, jap, kor
Dravidian	te, ta, ka
Other Settings	
Hungarian	
`lang2vec`	fin, tur, als
NLLB	fin, bul, hrv
Yoruba	
`lang2vec`	tha, ibo, fon
NLLB	ibo, hau, xho

Table 4: All languages used for the various experimental settings.

[5] https://huggingface.co/facebook/fasttext-language-identification

and Haspelmath, 2013), and Grambank (Skirgård et al., 2023). The `lang2vec` library (Littell et al., 2017) assembles typological feature vectors from different sources, enabling measurement of holistic typological distance between languages. The drawbacks of `lang2vec` include the lack of equal coverage of languages in terms of the number of features, and the coarseness of the linguistic properties (Ponti et al., 2019). To supplant the lack of coverage, we use `data2lang2vec` which uses textual data to inform feature-prediction of the missing features (Amirzadeh et al., 2025).

C.3 Data-driven Language Selection Methods

NLLB Language Embeddings Typological similarity between languages can also be derived in a data-driven manner, *e.g.*, from language embeddings. To derive the similarity between languages this way, we utilize the language embeddings of the distilled NLLB-200 model which implicitly contain much information with regards to the various languages especially with regards to morphosyntax and vocabulary.

Subword Evenness Introduced by Pelloni et al. (2022), Subword Evenness (SuE) is a metric quantifying the differences in the length of subword units for each individual language. It presents another data-driven approach to transfer language selection. In this metric, lower SuE scores have been demonstrated as a good predictor of transfer success to various low-resource languages. Thus, we select the 5 languages in the MADLAD dataset with the lowest SuE angles as calculated when using a BPE tokenizer, see Table 5.

Language Code	Language Code	Script	SuE Angle
≺azb	South Azerbaijani	Arabic	66.72
kat	Georgian	Georgian	72.57
zsm	Malay	Latin	79.80
≺kas	Kashmiri	Devanagari	80.79
rus	Russian	Cyrillic	81.21

Table 5: MADLAD Languages with the lowest SuE angles calculated using a BPE tokenizer. We note that MADLAD only has Norther Azerbaijani (Latin script), and not South Azerbaijani (Arabic script). Similarly, Kashmiri is only in MADLAD with the Arabic script, not with Devanagari.

C.4 Control Experiments with Unrelated Languages

To determine whether improvements over Creoles are truly the result of successful cross-lingual transfer, we compare the previously described methods against an equivalent adapter approach using entirely unrelated languages. We use three different sets of such unrelated languages. The first set is made up of the **Uralic languages** Finnish, Estonian, and Hungarian, which are written with the Latin script but have no phylogenetic relation to Haitian, Papiamento, and Sango. The second set consists of Chinese, Japanese, and Korean – so-called **CJK languages** –, which belong to three distinct language families and are written with three distinct scripts having little to no overlap with Atlantic and African Creole languages. The third group consists of the **Dravidian languages** Tamil, Telugu, and Kannada, which are closely related to each other but also written with three distinct scripts, and again are unrelated to the Creole languages in terms of phylogeny.

D Full Results

See Table 6 for our results translating between English and Creole languages using only 800 fine-tuning parallel segments.

E Training Loss

Table 7 compares loss when evaluating on a small subset of the NLLB CA-FT training data. Figure 3 indicates that the adapters we train learn successfully from the MADLAD-400 data on a selection from all language adapters.

F Native Speaker Evaluation

See Table 9 for a condensed categorization of translation errors under the Untrained Souping (📻), the IE Transfer (🌳), and the s and t Adapters ($s + t$) conditions.

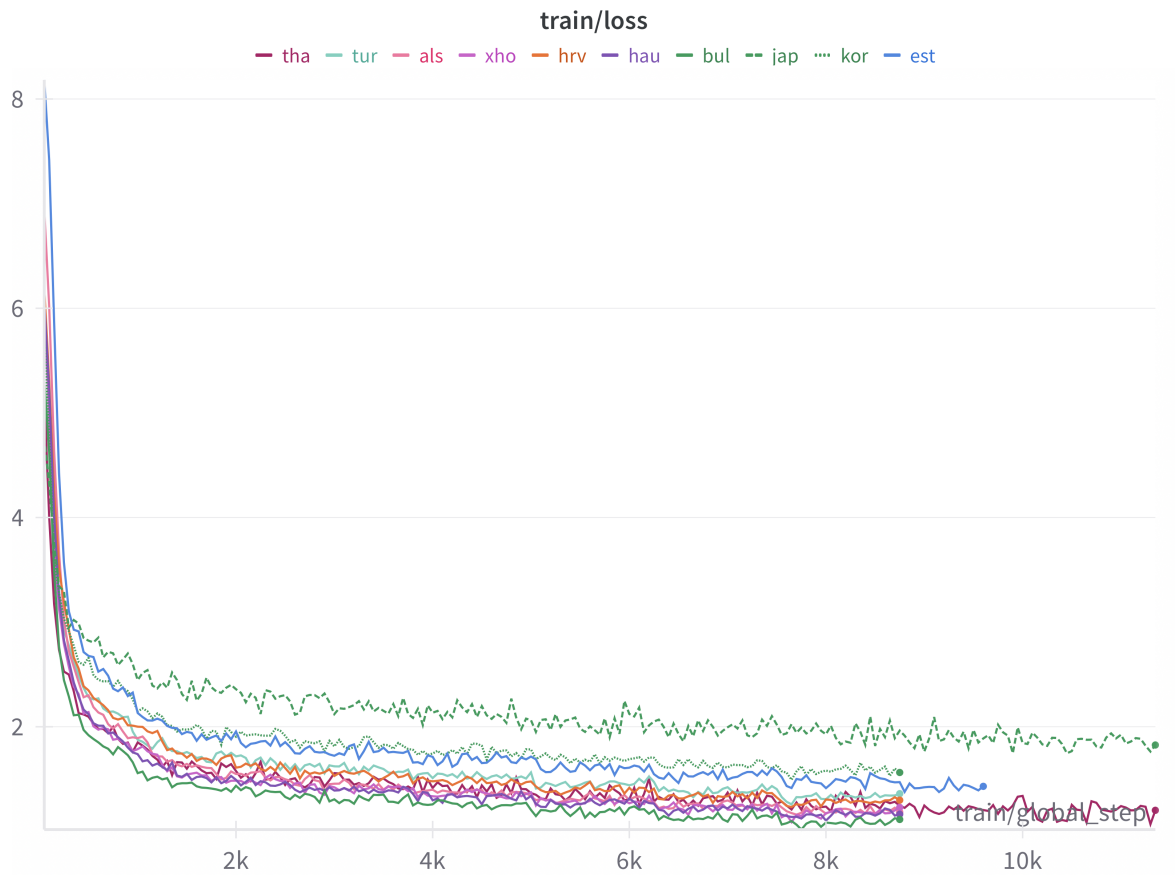


Figure 3: Training loss curves for a subset of adapter training experiments.

Experiments	en → hat	en → pap	en → sag	hat → en	pap → en	sag → en
Baselines						
dNLLB-200 w/ CA-FT	23.59	23.56	11.16	33.36	39.05	10.62
Language Adapters						
Source + Target LA	23.55	26.82	11.90	31.18	38.23	12.19
Phylogeny-Inspired Adaptation						
Target + IE Relatives	24.01	29.35	11.94	35.24	45.48	13.51
Target + NC Relatives	23.98	29.71	11.93	34.96	46.13	13.03
Typological Adaptation						
`lang2vec`	**24.28**	**30.21**	11.71	**35.63**	**46.95**	13.79
Target + Data-Driven (NLLB)	24.09	30.17	**12.01**	34.75	46.09	13.48
Miscellaneous						
Creole Transfer	23.99	29.87	11.71	34.24	44.75	13.48
Subword Evenness	24.23	29.73	11.80	35.20	46.20	**14.11**
Largest Improvement from dNLLB	+0.69	+6.65	+0.85	+2.27	+7.90	+3.49
Largest Improvement from Üstün	+0.73	+3.39	+0.11	+4.45	+8.63	+1.92
Control						
Uralic	24.31	30.08	11.44	**36.01**	46.36	**15.03**
CJK	23.95	29.18	11.60	35.64	46.91	14.71
Dravidian	24.21	29.17	11.69	35.51	46.66	14.45
Random (same init) (1 : 1)	24.21	28.89	11.86	36.07	46.03	14.79
Random (same init) (1 : 3)	24.10	29.65	11.55	37.10	46.66	15.39
Only random (same init)	-	-	-	36.84	45.01	14.90

Table 6: Mean BLEU scores on Flores-200 (after 1 run). Results follow adapter training defaults used in the Üstün et al. (2021) paper. Encoder and decoder adapters are trained together and trained further using cross-attention fine-tuning (CA-FT), and helper and target language adapters are averaged (souped) together. **These scores represent preliminary results, where significantly less data (800 segments) were used to do CA-FT, and thus these results do not correspond to Table 2.**

Experiment	hat → eng	pap → eng	sag → eng
s and t Adapters	0.594	1.106	1.908
Untrained s Adapter	**0.476**	**0.917**	**1.796**

Table 7: Validation loss when evaluating on the CA-FT training data. Averaging with the weights with the same initialization results in lower loss values.

Experiments	en → hu	en → yo	hu → en	yo → en
dNLLB-200	22.59	8.85	29.51	18.82
`lang2vec`	21.85	**8.51**	33.38	19.27
NLLB	21.75	8.45	33.31	19.14
SuE	**21.90**	8.37	33.47	19.56
Random (init) (1 : 3)	21.88	8.47	**33.97**	**20.70**
Difference	−0.69	−0.34	+4.46	+1.88

Table 8: Mean BLEU scores with our methodology, as applied to Yoruba and Hungarian. **These scores represent preliminary results, where significantly less data (800 segments) were used to do CA-FT, and thus not directly comparable to Table 2.**

Error	📻	🌳	$s+t$	Notes
Grammar Issues				
yo rann yo	✗	✗	✗	Aspectual issue: English translation is passive, Creole is a voluntary activity
te eksplore	✗	✗	✗	Aspectual error: English passive which also affects the sentence meaning
"many beautiful buildings to look at"	✓	✗	✗	🌳 and $s+t$ ungrammatical
"improve a considerable amount"	✓	✗	✗	🌳 and $s+t$ ungrammatical
"often provide some"	✓	✗	✓	🌳 ungrammatical
fatige mwens posib	✗	✗	✗	📻 omits it; 🌳 and $s+t$ to "fatigued less possibly"
konesans yo bay la	✗	✗	✗	to "they give/provide" with no clear subject (passive would be better)
mannyè efikas	✗	✗	✗	missing preposition "in" ("in efficient manner")
Translation Issues				
Lexical errors				
nimewotation	✗	✗	✗	$s+t$ wrongly translates it to "naming system"
benevola	✗	✗	✗	"travel benevola" or "benevola travel" (in fact "charity")
charabya	✗	✗	✗	Means "nonsense", "gibberish"
linets	✗	✗	✗	📻 to "lenses"; 🌳 to "links"; $s+t$ not translating
dyèl	✗	✓	✗	📻 to "deals"; 🌳 correctly; $s+t$ to "dyed"
patikilarite	✓	✓	✗	$s+t$ to "patikilarity"
kotidyen	✓	✓	✗	$s+t$ not translated
sètalò	✗	✗	✗	📻 to "sevental"; 🌳 to "most"; $s+t$ to "satellite" (in fact "sometimes")
bildinn	✗	✗	✗	📻 to "image"; 🌳 to "figurine"; $s+t$ to "picture" (in fact "building")
batiman	✗	✗	✗	📻 and 🌳 to "vessels"; $s+t$ to "baths" (in fact "building")
mannyè	✓	✗	✗	📻 correctly to "manner"; 🌳 and $s+t$ to "manny"
touch	✗	✗	✗	translated to "touch" though it means "(piano) key"
foumi militè	✗	✗	✗	📻 and 🌳 to "military foam"; $s+t$ to "military fountains" (in fact "army ants")
pyèj	✗	✗	✗	📻 to "hostage"; 🌳 and $s+t$ to "pigeon" (in fact "trap")
jako	✗	✗	✗	📻 and 🌳 to "jaco"; $s+t$ omitted/misinterpreted (in fact "parrot")
eskrimè	✗	✗	✗	📻 and $s+t$ to "skirmish"; 🌳 to "skater" (in fact "fencer")
parese	✗	✗	✗	to "parasites" (in fact "sloths")
pwomennen	✗	✗	✗	to "promoted" (in fact "walking")
eta espri	✓	✓	✗	📻 and 🌳 correctly to "state of mind"; $s+t$ to "spirit state"
Premye/Twazyèm Group	✗	✗	✗	to "First/Third Group" (in fact "First/Third Class")
mache	✗	✗	✗	to "walk" (in fact "migrate" in context)
Other errors				
kote pasaj enpótan yo	✗	✗	✗	📻 and 🌳 too literal; $s+t$ also adding "their" instead of "the"

Table 9: Condensed categorization errors under various translation conditions. 📻 refers to Untrained Souping, 🌳 indicates IE Transfer, and $s+t$ means s and t Adapters.

LLMs instead of Human Judges?
A Large Scale Empirical Study across 20 NLP Evaluation Tasks

Anna Bavaresco[1], Raffaella Bernardi[2], Leonardo Bertolazzi[2], Desmond Elliott[3], Raquel Fernández[1], Albert Gatt[4], Esam Ghaleb[5], Mario Giulianelli[6], Michael Hanna[1], Alexander Koller[7], André F. T. Martins[8], Philipp Mondorf[9], Vera Neplenbroek[1], Sandro Pezzelle[1], Barbara Plank[9], David Schlangen[10], Alessandro Suglia[11], Aditya K Surikuchi[1], Ece Takmaz[4], Alberto Testoni[12]

[1]University of Amsterdam, [2]University of Trento, [3]University of Copenhagen, [4]Utrecht University, [5]Max Planck Institute for Psycholinguistics, [6]ETH Zürich, [7]Saarland University, [8]Universidade de Lisboa & Unbabel, [9]LMU Munich & MCML, [10]University of Potsdam, [11]Heriot-Watt University [12]Amsterdam UMC
judgebench@googlegroups.com*

Abstract

There is an increasing trend towards evaluating NLP models with LLMs instead of human judgments, raising questions about the validity of these evaluations, as well as their reproducibility in the case of proprietary models. We provide JUDGE-BENCH, an extensible collection of 20 NLP datasets with human annotations covering a broad range of evaluated properties and types of data, and comprehensively evaluate 11 current LLMs, covering both open-weight and proprietary models, for their ability to replicate the annotations. Our evaluations show substantial variance across models and datasets. Models are reliable evaluators on some tasks, but overall display substantial variability depending on the property being evaluated, the expertise level of the human judges, and whether the language is human or model-generated. We conclude that LLMs should be carefully validated against human judgments before being used as evaluators.

https://github.com/dmg-illc/JUDGE-BENCH

1 Introduction

For many natural language processing (NLP) tasks, the most informative evaluation is to ask humans to judge the model output. Such judgments are traditionally collected in lab experiments or through crowdsourcing, with either expert or non-expert annotators, as illustrated in Fig. 1. Recently, there has been a trend towards replacing human judgments with automatic assessments obtained via large language models (LLMs; Chiang and Lee, 2023; Wang et al., 2023; Liu et al., 2023; Li et al., 2024; Zheng et al., 2024, *inter alia*). For example, the LLM could be instructed to rate a response generated by a dialogue system for its perceived plausibility, on a scale from 1 to 5. This drastically reduces the evaluation effort and is claimed to yield more reliable results across multiple evaluation rounds (Landwehr et al., 2023; Jiang et al., 2023b; Reiter, 2024; Dubois et al., 2024).

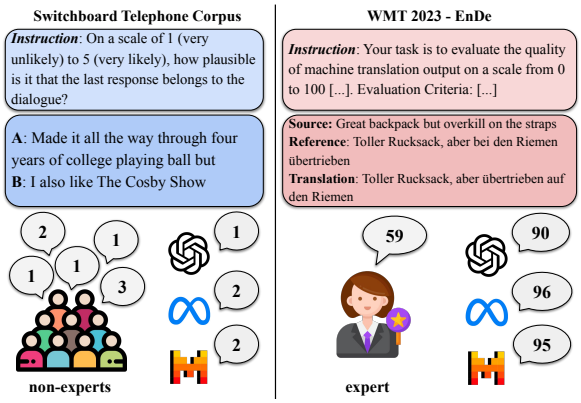


Figure 1: Evaluation by expert and non-expert human annotators and by LLMs for two tasks involving human-generated (left) and machine-generated text (right).

At the same time, the use of LLMs as judges of linguistic output raises new concerns: LLMs may be prone to errors or systematic biases that differ from those of humans, especially on subtle tasks such as evaluating toxicity, or reasoning. This may distort evaluation results and lead to incorrect conclusions. The problem is aggravated by explicit or implicit data leakage (Balloccu et al., 2024), which undermines the ability to make broad, generalisable claims beyond the single specific dataset under analysis. Specifically for closed models such as OpenAI's GPT series, there are serious reproducibility concerns, as LLMs may be retrained or retired at any time, making subsequent

*Authors listed in alphabetical order.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 238–255
July 27 - August 1, 2025

comparisons invalid or impossible.

Previous studies offer mixed evidence regarding the reliability of LLM evaluators. Some research concludes that they are effective, correlating well with human judgments (Liu et al., 2023; Zheng et al., 2024; Chen et al., 2023; Verga et al., 2024; Törnberg, 2023; Huang et al., 2024; Naismith et al., 2023; Gilardi et al., 2023; Kocmi and Federmann, 2023b), albeit with some caveats (Wang et al., 2023; Wu and Aji, 2025; Hada et al., 2024; Pavlovic and Poesio, 2024). In some cases, LLM evaluators can also provide pairwise preference judgments (Kim et al., 2024; Liusie et al., 2024; Liu et al., 2024a; Park et al., 2024; Tan et al., 2025), or fine-grained evaluation beyond a single score, such as error spans (Fernandes et al., 2023; Kocmi and Federmann, 2023a). In contrast, some studies highlight substantial biases in LLMs' behaviour as evaluators, both as compared against human judgments (Koo et al., 2024; Zeng et al., 2024; Baris Schlicht et al., 2024) and through intrinsic analyses (Wang et al., 2024; Liu et al., 2024b; Stureborg et al., 2024). These discrepancies likely stem from the limitations of this previous work, which typically relies on a few datasets and models, often restricted to closed-source proprietary models. The observation of such limitations has motivated recent work to develop finetuning methods for LLM judges designed to overcome certain biases (Zhu et al., 2025).

In this paper, we examine how well current LLMs can approximate human evaluators on a large scale. We prompt 11 among the most recent open-weight and proprietary LLMs to generate judgments on 20 datasets with human annotations on a wide range of quality dimensions, prompt styles, and tasks. Our evaluation goes beyond existing work by including a wide variety of *datasets* that differ in the *type of task* (e.g., translation, dialogue generation, etc.), the *property* being judged (e.g., coherence, fluency, etc.), the *type of judgments* (categorical or graded), and the *expertise of human annotators* (experts or non-experts). We provide JUDGE-BENCH, a benchmark which includes, upon release, a total of over 70,000 test instances with associated human judgments with an extensible codebase.

Our results indicate that LLMs align well with human judgments on certain tasks, like instruction following. However, their performance is *inconsistent* across and within annotation tasks. Elicitation methods like Chain-of-Thought prompting (Wei et al., 2022) do not reliably improve agreement, in line with recent findings (Sprague et al., 2025). Some proprietary models—in particular, GPT-4o—align better to humans, but there is a rather small gap with large open-source models, holding promise for the reproducibility of future evaluation efforts. Altogether, at the current stage of LLM development, we recommend validating LLM judges against task-specific human annotations before deploying them for any particular task.

2 Construction of JUDGE-BENCH

One key feature that differs across the datasets included in JUDGE-BENCH is the source of the data being evaluated, i.e., whether the items to be judged are generated by a model or produced by humans, as illustrated in Figure 1.

For model-generated items, the goal is to evaluate an NLP system. This includes both classic tasks such as machine translation or dialogue response generation, as well as less standard tasks for which automation has recently become an option thanks to LLMs, such as the generation of plans or logical arguments. For human-generated items, the goal is to assess properties of interest such as grammaticality or toxicity. This distinction allows us to understand whether LLMs have a positive bias towards machine-generated outputs—a tendency reported in prior work (Xu et al., 2024).

The datasets we consider cover a wide span of properties of interest, ranging from grammaticality and toxicity to coherence, factual consistency, and verbosity, *inter alia*. Many properties are relevant across multiple tasks (e.g., fluency and coherence), while others are more task-specific (e.g., the success of a generated plan or the correctness of a multi-step mathematical reasoning trace).

Our study focuses on English datasets or language pairs which include English as one of the languages. We keep track of whether the original annotation guidelines are available and whether the annotations are provided by experts or non-experts. We retain all available individual annotations. Dataset information is summarised in Table 2, Appendix A. All 20 datasets are formatted following a precise data schema to facilitate the integration of additional datasets. This makes JUDGE-BENCH easily extensible. We provide more details about the data schema in Appendix B.

	Dataset (# properties judged)	GPT-4o	Llama-3.1-70B	Mixtral-8x22B	Gemini-1.5	Mixtral-8x7B	Comm-R+	σ	*UB*
Categorical Annotations	CoLa (1)	0.34	0.46	0.54	0.45	**0.55**	0.12	0.16	-
	CoLa-grammar (63)	**0.47** ±0.22	0.28 ±0.24	0.28 ±0.23	0.26 ±0.24	0.21 ±0.18	0.13 ±0.14	0.14	-
	ToxicChat (2)	**0.49** ±0.36	0.41 ±0.26	0.45 ±0.27	0.45 ±0.35	0.36 ±0.12	0.28 ±0.35	0.1	-
	LLMBar-natural (1)	**0.84**	0.8	0.72	0.79	0.54	0.56	0.13	-
	LLMBar-adversarial (1)	**0.58**	0.46	0.2	0.29	0.06	0.11	0.2	-
	Persona Chat (2)	0.24 ±0.34	0.24 ±0.33	**0.58** ±0.59	-0.03 ±0.04	0.54 ±0.65	0.48 ±0.74	0.2	0.88
	Topical Chat (2)	**0.05** ±0.07	-0.02 ±0.02	-0.03 ±0.04	-0.03 ±0.04	0.02 ±0.03	0.01 ±0.02	0.07	0.58
	ROSCOE-GSM8K (2)	0.59 ±0.35	**0.64** ±0.27	0.62 ±0.38	0.6 ±0.24	0.58 ±0.36	0.0	0.15	-
	ROSCOE-eSNLI (2)	0.29 ±0.06	**0.38** ±0.08	0.13 ±0.13	0.11 ±0.18	0.1 ±0.11	0.03 ±0.05	0.14	-
	ROSCOE-DROP (2)	**0.29** ±0.08	0.27 ±0.07	0.2 ±0.12	0.08 ±0.05	0.13 ±0.21	0.03 ±0.04	0.13	-
	ROSCOE-CosmosQA (2)	0.16 ±0.07	**0.25** ±0.02	0.09 ±0.17	0.14 ±0.17	0.19 ±0.05	-0.03 ±0.01	0.1	-
	QAGS (1)	**0.72**	0.7	0.66	0.65	0.68	0.13	0.23	0.74
	Medical-safety (2)	0.01 ±0.03	-0.03 ±0.06	-0.02 ±0.09	-0.03 ±0.08	0.0 ±0.06	0.01 ±0.02	0.03	-
	DICES-990 (1)	-0.24	-0.17	-0.16	-0.12	-0.2	-0.09	0.05	0.27
	DICES-350-expert (1)	-0.2	-0.13	-0.15	-0.03	-0.11	0.01	0.08	-
	DICES-350-crowdsourced (1)	-0.22	-0.18	-0.08	-0.02	-0.11	-0.08	0.07	0.32
	Inferential strategies (1)	**0.42**	0.4	0.02	0.22	0.06	-0.02	0.19	1.0
	Average Cohen's κ	0.28 ±0.32	0.28 ±0.30	0.24 ±0.30	0.22 ±0.28	0.21 ±0.28	0.10 ±0.18		
Graded Annotations	Dailydialog (1)	**0.69**	0.6	0.55	0.63	0.63	0.52	0.06	0.79
	Switchboard (1)	**0.66**	0.45	0.63	0.59	0.56	0.36	0.11	0.8
	Persona Chat (4)	**0.22** ±0.11	-0.02 ±0.2	0.16 ±0.1	0.1 ±0.09	0.02 ±0.15	0.07 ±0.13	0.2	0.61
	Topical Chat (4)	0.26 ±0.03	**0.28** ±0.1	0.13 ±0.04	0.17 ±0.12	0.21 ±0.18	0.14 ±0.05	0.07	0.56
	Recipe-generation (6)	**0.78** ±0.05	0.66 ±0.07	0.6 ±0.15	0.67 ±0.09	0.57 ±0.24	0.32 ±0.28	0.18	0.65
	ROSCOE-GSM8K (2)	0.82 ±0.12	**0.83** ±0.11	0.81 ±0.14	0.81 ±0.12	0.79 ±0.13	0.68 ±0.2	0.15	-
	ROSCOE-eSNLI (2)	**0.49** ±0.24	0.4 ±0.16	0.38 ±0.17	0.35 ±0.21	0.32 ±0.12	0.09 ±0.08	0.14	-
	ROSCOE-DROP (2)	0.57 ±0.22	**0.59** ±0.16	0.44 ±0.15	0.44 ±0.13	0.32 ±0.12	0.21 ±0.22	0.13	-
	ROSCOE-CosmosQA (2)	**0.57** ±0.18	0.55 ±0.18	0.51 ±0.16	**0.57** ±0.17	0.53 ±0.21	0.33 ±0.25	0.1	-
	NewsRoom (4)	**0.59** ±0.02	**0.59** ±0.03	0.44 ±0.05	0.55 ±0.03	0.5 ±0.07	0.36 ±0.06	0.1	0.62
	SummEval (4)	0.35 ±0.06	0.44 ±0.14	**0.54** ±0.08	0.38 ±0.02	0.48 ±0.02	0.19 ±0.06	0.13	-
	WMT 2020 En-De (1)	**0.63**	0.37	0.51	0.46	0.2	0.42	0.15	0.81
	WMT 2020 Zh-En (1)	**0.54**	0.39	0.48	0.41	0.25	0.42	0.1	0.62
	WMT 2023 En-De (1)	0.22	0.14	**0.23**	0.16	0.17	0.22	0.04	-
	WMT 2023 Zh-En (1)	0.17	0.14	**0.19**	0.14	0.15	0.15	0.02	-
	Average Spearman's ρ	0.50 ±0.21	0.43 ±0.22	0.44 ±0.19	0.43 ±0.21	0.38 ±0.22	0.30 ±0.17		

Table 1: Scores per dataset for the models with ≥98% valid response rates (results for all models in Tab. 6, App. H): Cohen's kappa for categorical annotations and Spearman's correlation for graded annotations. Boldface marks best model performance per dataset. Spearman's correlations are generally significant ($p < 0.05$), with the exception of the Persona Chat and Topical Chat datasets (see Tab. 6 in Appendix H for more details). Datasets with both categorical and graded annotations appear twice. Datasets in blue concern human-generated language, while those in red concern model-generated text. 'σ' denotes the standard deviation of the scores across models per dataset (averaged over properties if more than one is judged per dataset). Upper-bound estimates (*UB*) indicate the agreement between individual and aggregated human judgments.

3 Model Selection and Experiment Design

Models. We select representative proprietary and open-weight models of various sizes that show high performance across several tasks on the Open LLM and Chatbot Arena Leaderboards (Chiang et al., 2024): GPT-4o (OpenAI, 2024), LLaMA-3.1 (8B and 70B; AI@Meta 2024), Gemini-1.5 (Reid et al., 2024), Mixtral (8x7B and 8x22B; Jiang et al. 2024), Command R and Command R+ (Cohere and Cohere for AI, 2024a,b), OLMo (Groeneveld et al., 2024), Starling-7B (Zhu et al., 2024), and Mistral (Jiang et al., 2023a). See Appendix E for inference procedure details.

Prompts. Since most datasets include the original instructions used to gather human judgments, we use these instructions directly as prompts for the model, with additional guidelines to constrain the models' output and minimise verbosity: '*Answer with one of {}. Do not explain your answer.*' When the original instruction for collecting human judgments is unavailable, we create a prompt based on relevant information from the original paper, such as the task description and the definitions of the evaluation metrics. We also experiment with alternative prompting strategies, including Chain-of-Thought, few-shot prompts, and prompt paraphrases. However, none of these strategies leads to systematic improvements. See Appendix H for full details and results. All prompts are provided in the codebase.

Evaluation. Models do not always respond to the prompts as requested (e.g., they may refuse to

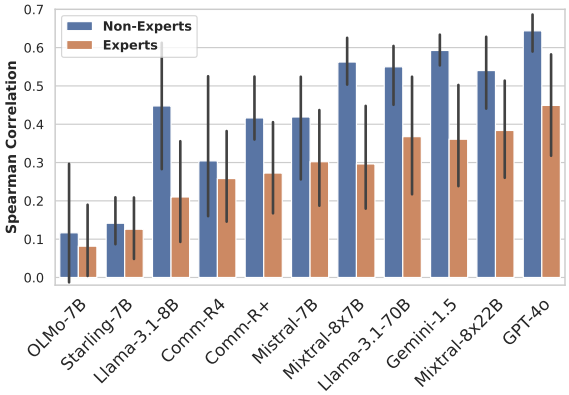


Figure 2: Average model correlation with human experts vs. non-experts in datasets with graded annotations.

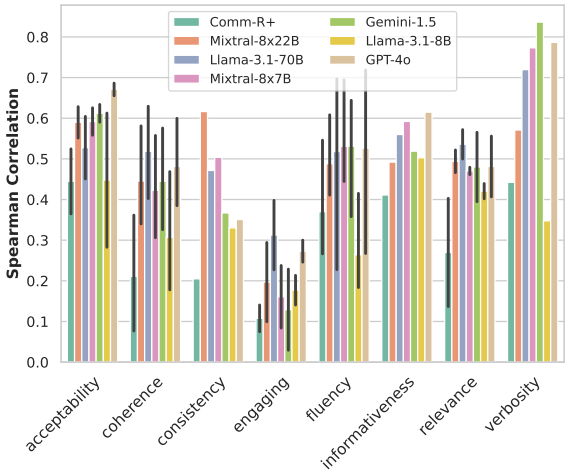


Figure 3: Correlation for properties with graded judgments. Averages and error bars when the property is present in more than one dataset.

answer if they perceive the prompt as sensitive). We therefore use the following evaluation protocol:

- To obtain the same number of judgments across models for a given dataset, we replace invalid LLM responses with judgments randomly sampled from the relevant set of categorical or graded annotations. Figure 5 in Appendix F shows the rate of valid responses per model.
- Graded annotations, such as in WMT 2020 (Freitag et al., 2021), assess language quality on a continuous scale (e.g., a score from 0 to 100, or Likert-scale ratings), capturing varying degrees of fluency, adequacy, or overall translation quality; whereas categorical annotations, like those in CoLa (Warstadt et al., 2019), involve binary judgments (e.g., grammatically acceptable or not). For the former, we compute Spearman's correlation (ρ) between model and human judgments; for the latter, we compute Cohen's κ.
- When multiple individual human judgments are available (typically three, see Table 2 in Appendix A), we estimate an upper bound by computing the average Spearman's ρ or Cohen's κ between bootstrapped single-rater responses and the aggregated responses across raters. Appendix C provides details on the upper bounds.

4 Results

Scores vary substantially across models. For any given model, they vary both across datasets and properties being judged. Table 1 presents detailed results for the 6 models that exhibit the largest rate of valid responses ($\geq$98%). GPT-4o ranks first across several evaluation scenarios, but the Llama-3.1-70B and Mixtral-8x22B open models are relatively close and outperform GPT-4o on some assessment types, such as categorical sentence acceptability (CoLa) and graded summary quality (SummEval). Overall, the high degree of variability is not fully accounted for by the inherent difficulty of the annotation tasks, as reflected in the human upper bound. Moreover, except for a few datasets (e.g., QAGS, Recipe-generation, and NewsRoom), model scores remain notably below the upper bound.

Among the property types with the lowest human-model alignment are toxicity and safety (in particular on DICES and Medical-safety), where model scores can be even negative and valid response rates particularly low (see Fig. 6 in Appendix F). This is due in part to the guardrails associated with these tasks (Weidinger et al., 2023). We find that, especially in the medical domain, many models tend to provide explanations instead of producing a judgment (see Appendix G).

Despite the high variability across models and datasets, we observe several notable trends. For graded annotations (Fig. 2), all models achieve higher correlations with annotations by non-expert human judges compared to expert annotators, echoing recent findings by Aguda et al. (2024). One possible explanation is that non-experts might rely on surface-level features, which could align more closely with the patterns LLMs are most attuned to, while experts apply stricter, domain-specific criteria. This remains speculative and calls for further investigation.

Figure 3 shows correlation results across different datasets for the subset of properties that exclusively have graded judgments. When applicable,

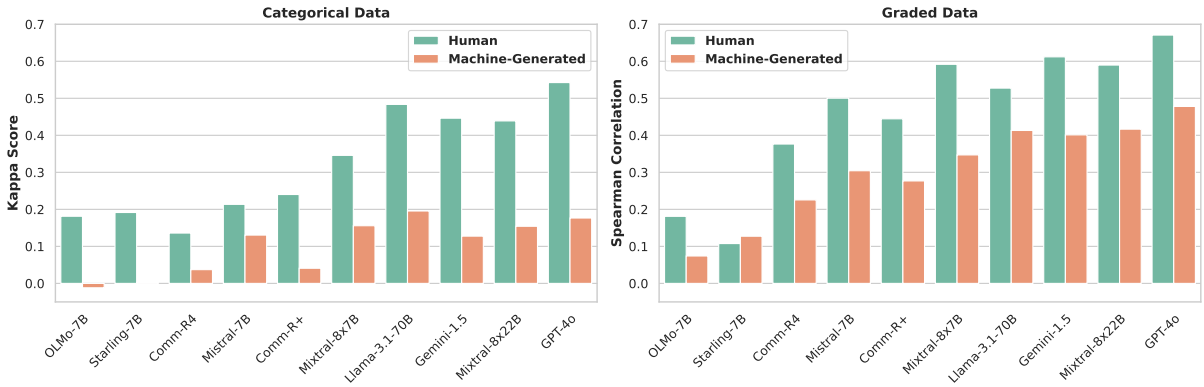


Figure 4: Scores (Cohen's κ for categorical annotations and Spearman's correlation for graded annotations) on test items involving human language vs. machine-generated outputs.

we average results across datasets including annotations for the same property. We provide more details about these properties in Appendix D. The proprietary models GPT-4o and Gemini-1.5 exhibit the highest scores when evaluating acceptability and verbosity, while the two Mixtral open models show the strongest correlations for coherence and consistency. Correlation with the engagingness property remains consistently low across all models. Overall, *no single model demonstrates a clear superiority* over others across all properties; instead, different quality dimensions are better assessed by different models. This calls into question the widespread practice of using a single model – typically a proprietary one like those from the GPT family—to evaluate a diverse range of linguistic properties.

Finally, as shown in Figure 4, all models achieve better alignment with human judgments when evaluating human language than when assessing machine-generated text, both for categorical and graded annotations. This result aligns with the findings by Xu et al. (2024), suggesting that LLMs display a bias towards their own generation. More broadly, this trend calls for caution when using LLMs to automatically evaluate the output of NLP systems.

5 Conclusions

In response to current trends in evaluation, in this paper we conducted a large-scale study of the correlation between human and LLM judgments across 20 datasets, considering factors such as the properties being assessed, the expertise level of the human judges, and whether the data is model- or human-generated. On some tasks, such as instruction following and the generation of mathematical reasoning traces, models can be reliably used as evaluators. Overall, however, models' agreement with human judgments varies widely across datasets, evaluated properties, and data sources; and depends on the level of expertise of human judges. Furthermore, elicitation strategies such as Chain-of-Thought prompting do not consistently improve agreement levels, in line with recent findings (Sprague et al., 2025). We recommend validation and calibration of LLMs against task-specific human judgments prior to their deployment as evaluators. To facilitate this process, we release JUDGEBENCH, a benchmark that enables systematic evaluation across a diverse range of tasks and is easily extensible to include any new task of interest.

Limitations

One limitation of the experimental design of our work is that correlation with human judges may not be the most appropriate way to validate LLM evaluators. Indeed, there may be domains where human annotators and LLM evaluators appear aligned simply because they are affected by similar biases. Therefore, depending on the task at hand, it may be necessary to validate the reliability of human annotators as well.

Another limitation concerns the use of existing tasks and datasets without reassessing their quality or representativeness of actual downstream tasks. While we did our best to select a wide set of tasks meaningful to the NLP community, we acknowledge that these tasks could not be equally meaningful for end-users, and that employing existing datasets could arguably lead to potential risks and shortcomings, such as data leakage.

In contrast to approaches that use LLMs for pairwise preference evaluation, e.g., PairEval (Park et al., 2024) or JudgeBench (Tan et al., 2025),[1] this paper focuses on evaluating the performance of LLMs on generating judgements for categorical and graded responses. We leave the extension of JUDGE-BENCH to include pairwise preference evaluation and other recent evaluation methods, such as Prometheus 2 (Kim et al., 2024), for future work.

Finally, our work mostly focuses on English-language datasets—with the exception of datasets focussing specifically on machine-translation outputs. It remains to be seen whether LLMs' meta-evaluation abilities vary across different languages.

Acknowledgements

This work emerged from discussions at a workshop organised by Raquel Fernández and Sandro Pezzelle at MFO, the Oberwolfach Research Institute for Mathematics in the German Black Forest, on behalf of the ELLIS NLP programme. The event was funded by the state of Baden-Württemberg (Germany) and organised in collaboration with the ELLIS Institute Tübingen and the Max Planck Institute for Intelligent Systems. We furthermore acknowledge our funders. In particular, AB, RF, and AT were supported by the European Research Council (ERC Consolidator Grant DREAM 819455 to RF), as well as BP (ERC Consolidator Grant DIALECT 101043235 to BP). DE was supported by a research grant (VIL53122) from VILLUM FONDEN. EG was supported by the Dutch Research Council (Gravitation grant 024.001.006 to the Language in Interaction consortium). MG was supported by an ETH Zurich Postdoctoral Fellowship. MH was supported in part by an OpenAI Superalignment Fellowship. AM was supported by the European Research Council (DECOLLAGE, ERC-2022-CoG 101088763) and by Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020. We acknowledge ISCRA for awarding this project access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CINECA (Italy).

[1]Some time after an early version of this paper became available as a pre-print, accompanied by our JudgeBench code, the independent work by Tan et al. (2025) appeared, which describes a benchmark that the authors named JudgeBench. This name clash is unfortunate, but since in the meantime our paper has seen some uptake, we have decided against trying to resolve it.

References

Gavin Abercrombie and Verena Rieser. 2022. Risk-graded safety for handling medical queries in conversational AI. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 234–243, Online only. Association for Computational Linguistics.

Toyin D. Aguda, Suchetha Siddagangappa, Elena Kochkina, Simerjot Kaur, Dongsheng Wang, and Charese Smiley. 2024. Large language models as financial data annotators: A study on effectiveness and efficiency. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10124–10145, Torino, Italia. ELRA and ICCL.

AI@Meta. 2024. Llama 3.1 model card.

Lora Aroyo, Alex Taylor, Mark Díaz, Christopher Homan, Alicia Parrish, Gregory Serapio-García, Vinodkumar Prabhakaran, and Ding Wang. 2023. Dices dataset: Diversity in conversational ai evaluation for safety. In *Advances in Neural Information Processing Systems*, volume 36, pages 53330–53342. Curran Associates, Inc.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93, St. Julian's, Malta. Association for Computational Linguistics.

Ipek Baris Schlicht, Defne Altiok, Maryanne Taouk, and Lucie Flek. 2024. Pitfalls of conversational LLMs on news debiasing. In *Proceedings of the First Workshop on Language-driven Deliberation Technology (DELITE) @ LREC-COLING 2024*, pages 33–38, Torino, Italia. ELRA and ICCL.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. 2023. Exploring the use of large language models for reference-free text quality evaluation: An empirical study. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 361–374, Nusa Dua, Bali. Association for Computational Linguistics.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: an open platform for evaluating llms by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

Cohere and Cohere for AI. 2024a. Command R Model Card.

Cohere and Cohere for AI. 2024b. Command R+ Model Card.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André FT Martins, Graham Neubig, Ankush Garg, Jonathan H Clark, Markus Freitag, and Orhan Firat. 2023. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1066–1083.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Acoustics, speech, and signal processing, ieee international conference on*, volume 1, pages 517–520. IEEE Computer Society.

Olga Golovneva, Moya Peng Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tur. 2019. Topical-chat: Towards knowledge-grounded open-domain conversations. In *Proc. Interspeech 2019*, pages 1891–1895.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. 2024. Olmo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Rishav Hada, Varun Gumma, Adrian de Wynter, Harshita Diddee, Mohamed Ahmed, Monojit Choudhury, Kalika Bali, and Sunayana Sitaram. 2024. Are large language model-based evaluators the solution to scaling up multilingual evaluation? In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1051–1070, St. Julian's, Malta. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Fan Huang, Haewoon Kwak, Kunwoo Park, and Jisun An. 2024. ChatGPT rates natural language explanation quality like humans: But on which scales? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3111–3132, Torino, Italia. ELRA and ICCL.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on*

Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023b. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, Toronto, Canada. Association for Computational Linguistics.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Masaaki Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, Mariya Shmatova, and Jun Suzuki. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023a. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775, Singapore. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023b. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2024. Benchmarking cognitive biases in large language models as evaluators. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 517–545, Bangkok, Thailand. Association for Computational Linguistics.

Fabian Landwehr, Erika Varis Doggett, and Romann M. Weber. 2023. Memories for virtual AI characters. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 237–252, Prague, Czechia. Association for Computational Linguistics.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, Yuxuan Lai, Chongyang Tao, and Shuai Ma. 2024. Leveraging large language models for NLG evaluation: Advances and challenges. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16028–16045, Miami, Florida, USA. Association for Computational Linguistics.

Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. 2023. ToxicChat: Unveiling hidden challenges of toxicity detection in real-world user-AI conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4694–4702, Singapore. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2024a. Aligning with human judgement: The role of pairwise preference in large language model evaluators. In *First Conference on Language Modeling*.

Yiqi Liu, Nafise Moosavi, and Chenghua Lin. 2024b. LLMs as narcissistic evaluators: When ego inflates evaluation scores. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12688–12701, Bangkok, Thailand. Association for Computational Linguistics.

Adian Liusie, Potsawee Manakul, and Mark Gales. 2024. LLM comparative assessment: Zero-shot NLG evaluation through pairwise comparisons using large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 139–151, St. Julian's, Malta. Association for Computational Linguistics.

Shikib Mehri and Maxine Eskenazi. 2020. USR: An unsupervised and reference free evaluation metric for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 681–707, Online. Association for Computational Linguistics.

Philipp Mondorf and Barbara Plank. 2024. Comparing inferential strategies of humans and large language models in deductive reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9370–9402, Bangkok, Thailand. Association for Computational Linguistics.

Ben Naismith, Phoebe Mulcaire, and Jill Burstein. 2023. Automated evaluation of written discourse coherence using GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 394–403, Toronto, Canada. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Hamed Nili, Cai Wingfield, Alexander Walther, Li Su, William Marslen-Wilson, and Nikolaus Kriegeskorte. 2014. A toolbox for representational similarity analysis. *PLoS computational biology*, 10(4):e1003553.

OpenAI. 2024. Gpt-4o model card.

ChaeHun Park, Minseok Choi, Dohyun Lee, and Jaegul Choo. 2024. PairEval: Open-domain dialogue evaluation with pairwise comparison. In *First Conference on Language Modeling*.

Maja Pavlovic and Massimo Poesio. 2024. The effectiveness of LLMs as annotators: A comparative overview and empirical analysis of direct representation. In *Proceedings of the 3rd Workshop on Perspectivist Approaches to NLP (NLPerspectives) @ LREC-COLING 2024*, pages 100–110, Torino, Italia. ELRA and ICCL.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Ehud Reiter. 2024. Can LLM-based eval replace human evaluation? Blog post.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2025. To CoT or not to CoT? Chain-of-thought helps mainly on math and symbolic reasoning. In *The Thirteenth International Conference on Learning Representations*.

Katharina Stein, Lucia Donatelli, and Alexander Koller. 2023. From sentence to action: Splitting AMR graphs for recipe instructions. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations*, pages 52–67, Nancy, France. Association for Computational Linguistics.

Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *Preprint*, arXiv:2405.01724.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y. Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. 2025. JudgeBench: A benchmark for evaluating LLM-based judges. In *The Thirteenth International Conference on Learning Representations*.

Petter Törnberg. 2023. ChatGPT-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning. *arXiv preprint arXiv:2304.06588*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment. *Preprint*, arXiv:2310.16944.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models. *arXiv preprint arXiv:2404.18796*.

Sarenne Carrol Wallbridge, Catherine Lai, and Peter Bell. 2022. Investigating perception of spoken dialogue acceptability through surprisal. In *Proc. Interspeech 2022*, pages 4506–4510.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is ChatGPT a good NLG evaluator? a preliminary study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 1–11, Singapore. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Alex Warstadt and Samuel R. Bowman. 2020. Linguistic analysis of pretrained sentence encoders with acceptability judgments. *Preprint*, arXiv:1901.03438.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Laura Weidinger, Maribeth Rauh, Nahema Marchal, Arianna Manzini, Lisa Anne Hendricks, Juan Mateos-Garcia, Stevie Bergman, Jackie Kay, Conor Griffin, Ben Bariach, et al. 2023. Sociotechnical safety evaluation of generative ai systems. *arXiv preprint arXiv:2310.11986*.

Minghao Wu and Alham Fikri Aji. 2025. Style over substance: Evaluation biases for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 297–312, Abu Dhabi, UAE. Association for Computational Linguistics.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. 2024. Pride and Prejudice: LLM Amplifies Self-Bias in Self-Refinement. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15474–15492, Bangkok, Thailand. Association for Computational Linguistics.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024. Evaluating large language models at evaluating instruction following. In *The Twelfth International Conference on Learning Representations*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems*, 36.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. 2024. Starling-7B: Improving LLM helpfulness & harmlessness with RLAIF. In *First Conference on Language Modeling*.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2025. JudgeLM: Fine-tuned Large Language Models are Scalable Judges. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*. arXiv. ArXiv:2310.17631 [cs].

Appendix

A Datasets

This section provides brief descriptions of the datasets employed in our study. Table 2 summarises relevant dataset information. Note that dataset sizes as reported in Table 2 refer to the number of annotated samples (not to the total number of collected annotations) and might therefore differ from the figures reported in the original papers. Table 3 reports Krippendorf's α for those datasets with multiple public human annotations.

CoLa (Warstadt et al., 2019). The Corpus of Linguistic Acceptability (CoLA) consists of 10657 sentences from 23 linguistics publications, expertly annotated for acceptability (grammaticality) by their original authors.

CoLa-grammar (Warstadt and Bowman, 2020). The dataset consists of a grammatically annotated version of the CoLA development set. Each sentence in the CoLA development set is labelled with boolean features indicating the presence or absence of a particular grammatical construction (usually syntactic in nature). Two related sets of features are considered: 63 minor features correspond to fine-grained phenomena, and 15 major features correspond to broad classes of phenomena.

ToxicChat (Lin et al., 2023). collect binary judgments on the toxicity and 'jailbreaking' nature (prompt hacks deliberately intended to bypass safety policies and induce models to generate unsafe content) of human prompts to LLMs. While the original dataset contains a mix of human- and automatically-annotated instances, here we only consider the human-annotated prompts.

LLMBar (Zeng et al., 2024). LLMBar is a dataset targeted at evaluating the instruction-following abilities of LLMs. Each entry of this dataset consists of an instruction paired with two different outputs, one correctly following the instruction and the other deviating from it. LLMBar has an `adversarial` split where deviating outputs are carefully constructed to 'fool' LLM-based evaluators and a `natural` split where deviating outputs are more naturalistic.

Dataset	Task	Size	# Annot.	Type	Guidelines	Expert	Leaked
CoLA (Warstadt et al., 2019)	Acceptability	1,043	-	Categorical	✗	✓	✓
CoLA-grammar (Warstadt and Bowman, 2020)	Acceptability	1,043	-	Categorical	✗	✓	✓
Switchboard (Wallbridge et al., 2022)	Acceptability	100	3-6	Graded	✓	✗	
Dailydialog (Wallbridge et al., 2022)	Acceptability	100	3-6	Graded	✓	✗	
Inferential strategies (Mondorf and Plank, 2024)	Reasoning	300	-	Categorical	✓	✓	✗
ROSCOE (Golovneva et al., 2023)	Reasoning	756	-	Categorical + Graded	✓	✓	
Recipe-generation (Stein et al., 2023)	Planning	52	-	Graded	✓		
Medical-safety (Abercrombie and Rieser, 2022)	Toxicity & Safety	3,701	-	Preference	✓	✓	
DICES (Aroyo et al., 2023)	Toxicity & Safety	1,340	~70 + ~120	Categorical	✗	Mixed	
ToxicChat (Lin et al., 2023)	Toxicity & Safety	5,654	-	Categorical	✗	✓	
Topical Chat (Mehri and Eskenazi, 2020)	Dialogue	60	3	Graded + Categorical	✗	✓	
Persona Chat (Mehri and Eskenazi, 2020)	Dialogue	60	3	Graded + Categorical	✗	✓	
WMT 2020 En-De (Freitag et al., 2021)	Machine Translation	14,122	3	Graded	✗	✓	
WMT 2020 Zh-En (Freitag et al., 2021)	Machine Translation	19,974	3	Graded	✗	✓	
WMT 2023 En-De (Kocmi et al., 2023)	Machine Translation	6,588	-	Graded	✗	✓	
WMT 2023 Zh-En (Kocmi et al., 2023)	Machine Translation	13,245	-	Graded	✗	✓	
G-Eval / SummEval (Liu et al., 2023)	Summarisation	1,600	-	Graded	✓		✓
QAGS (Wang et al., 2020)	Summarisation	953	3	Categorical	✓	✗	
NewsRoom (Grusky et al., 2018)	Summarisation	420	3	Graded	✓	✗	✓
LLMBar (Zeng et al., 2024)	Instruction Following	419	-	Categorical	✓	✓	✗

Table 2: Overview of the main features of the datasets considered in the study. Note that 'Size' refers to the number of annotated samples, not to the total number of human annotations. '# Annot.' refers to the number of available individual annotations, if any, which we use to estimate the human upper bound. Note that datasets with only a single annotation per sample, or which only report the average over multiple annotations are not included in '# Annot.'. Information on possible data leakage was retrieved from Balloccu et al. (2024).

Topical Chat and Persona Chat (Mehri and Eskenazi, 2020). These datasets contain human judgments on the quality of machine- and human-generated responses based on the provided dialogue context. The annotated dialogues were selected from Topical Chat (Gopalakrishnan et al., 2019)—a dataset collecting human-human conversations on provided facts—and Persona Chat (Zhang et al., 2018), which contains human-human persona-conditioned conversations. Each response is evaluated on 6 attributes: Understandable, Natural, Maintains Context, Interesting/Engaging, Uses Knowledge, and Overall Quality.

ROSCOE (Golovneva et al., 2023). collect human judgments assessing the quality of GPT-3's reasonings. The output reasonings are elicited by inputting GPT-3 with questions selected from 4 commonly used reasoning datasets, i.e., CosmosQA (Huang et al., 2019), DROP (Dua et al., 2019), e-SNLI (Camburu et al., 2018) and GSM8K (Cobbe et al., 2021). While ROSCOE provides annotations on each step of the reasoning trace, here we only consider the global judgments over the whole reasoning.

QAGS (Wang et al., 2020). QAGS consists of annotations judging the factual consistency of one-sentence model-generated summaries of news articles. The gold-standard summaries and articles are collected from CNN/DailyMail (Hermann et al., 2015) and XSUM (Narayan et al., 2018).

Medical-safety (Abercrombie and Rieser, 2022). This dataset consists of 3701 pairs of medical queries (collected from a subreddit on medical advice) and both machine-generated and human-generated answers. Queries were classified by human annotators according to their severity (from 'Not medical' to 'Serious', with 'Serious' indicating that emergency care would be required) and answers were categorised based on their risk level (from 'Non-medical' to 'Diagnosis/Treatment').

DICES (Aroyo et al., 2023). The DICES datasets consist of a series of machine-generated responses whose safety is judged based on the previous conversation turns (context). While the original dataset provides fine-grained annotations with answers to questions targeting specific aspects of safety, here we only consider the 'overall' categorisation comprehensive of all aspects. In `DICES 990` safety is judged by crowdsourced annotators, whereas in `DICES 350` both expert and crowdsourced annotations are provided.

Inferential strategies (Mondorf and Plank, 2024). This dataset contains annotations on the logical validity of reasoning steps that models—in this case, `Llama-2-chat-hf3` (Touvron et al., 2023), `Mistral-7B-Instruct-v0.2` (Jiang et al., 2023a) and `Zephyr-7b-beta` (Tunstall et al., 2023)—generate when prompted to solve problems

of propositional logic. Binary labels are assigned to each response, indicating whether the rationale provided by the model is sound (True) or not (False). Each model is assessed on 12 problems of propositional logic across 5 random seeds, resulting in a total of 60 responses per model.

Switchboard and Dailydialog (Wallbridge et al., 2022). Switchboard includes acceptability judgments collected using stimuli from the Switchboard Telephone Corpus (Godfrey et al., 1992). More specifically, the judgments refer to how plausible it is that a specific response belongs to a telephonic dialogue. The same kind of judgments are provided for Dailydialog, which collects written dialogues intended to mimic conversations that could happen in real life.

Recipe-generation (Stein et al., 2023). This dataset contains human annotations assessing the quality of machine-generated recipes based on 6 attributes: grammar, fluency, verbosity, structure, success, overall.

NewsRoom (Grusky et al., 2018). This dataset includes human judgments on the quality of system-generated summaries of news articles. More specifically, annotators evaluated summaries across two semantic dimensions (informativeness and relevancy) and two syntactic dimensions (fluency and coherence).

SummEval and G-Eval (Fabbri et al., 2021; Liu et al., 2023). These datasets include summaries generated by multiple recent summarisation models trained on the CNN/DailyMail dataset (Hermann et al., 2015). Summaries are annotated by both expert judges and crowdsourced workers on 4 dimensions: coherence, consistency, fluency, relevance.

WMT 2020 En-De and Zh-En (Freitag et al., 2021). These datasets are a re-annotated version of the English-to-German and Chinese-to-English test sets taken from the WMT 2020 news translation task. The annotation was carried out by raters who are professional translators and native speakers of the target language using a Scalar Quality Metric (SQM) evaluation on a 0–6 rating scale.

WMT 2023 En-De and Zh-En (Kocmi et al., 2023). These datasets are the English-to-German and Chinese-to-English test sets taken from the General Machine Translation Task organised as part of the 2023 Conference on Machine Translation (WMT). In contrast to previous editions, the evaluation of translation quality was conducted by a professional or semi-professional annotator pool rather than utilising annotations from MTurk. Annotators were asked to provide a score between 0 and 100 on a sliding scale.

	Dataset	Krippendorf's α
Categorical	Topical Chat	0.08
	QAGS	0.49
	DICES-990	0.14
	DICES-350-crowdsourced	0.16
	Persona Chat	0.33
	Inferential strategies	1.0
Graded	Dailydialog	0.59
	Switchboard	0.57
	Persona Chat	0.33
	Topical Chat	0.08
	Recipe-generation	0.41
	NewsRoom	0.11
	WMT 2020 En-De	0.5
	WMT 2020 Zh-En	0.09

Table 3: Inter-rater agreement for datasets with multiple human annotations. Datasets in blue concern human-generated language, while those in red concern model-generated text.

B The JUDGE-BENCH Data Schema

To facilitate extending our benchmark, we adopt a shared schema used to pre-process all datasets. Our publicly available code base includes an example[2] of this format as well as instructions on how to verify that newly added datasets comply with it.

The Json-based JUDGE-BENCH data schema ensures that the following fields are included for each dataset:

- `dataset`: the name of the dataset;
- `dataset_url`: the URL where the original dataset can be downloaded, as opensourced by their creators;
- `annotations`: an overview of the properties annotated for each dataset, along with information on how they are measured and prompt-like instructions similar to those originally given to the human annotators (when applicable);
- `instances`: dataset instances including the piece of text to be judged, aggregated human

[2] https://github.com/dmg-illc/JUDGE-BENCH/blob/master/data/example.json

judgments and, when available, individual human annotations.

We note that, while we do not systematically explore inter-annotator variations at the instance level, the data schema we adopt allows for conducting this type of analysis in future work.

C Upper Bound Estimation for Model Correlations

Whenever multiple human annotations were publicly available for a property (see Table 3 for inter-annotator agreement scores), we computed upper-bound estimates for the correlations achievable by models. The intuition behind these estimates, borrowed from neuroscience (Nili et al., 2014), is that the maximum correlation a model can achieve with aggregated human responses is bounded by the average correlation between single-participant responses and the aggregated responses across participants. We applied a similar logic to the human judgments used in the present study and combined it with a bootstrapping approach. For each annotated property, we bootstrapped single-participant responses by sampling 1000 times from the available human responses, excluding data points where a single annotation was available. Next, we computed the alignment between each of the bootstrapped-participant arrays and the array of aggregated responses. Alignment was computed as Spearman's correlation for graded judgments and Cohen's kappa for categorical judgments. Finally, we estimated the upper bound as the average of the 1000 alignment measures. In cases where alignment between bootstrapped and aggregated responses could not be computed—because the variance of the bootstrapped responses was null—values were replaced with an average of the 'non-nan' correlations.

We emphasise that these upper bounds are estimates and, as such, are subject to errors. Therefore, it may happen that model performance exceeds these upper bounds.

D Properties with Graded Judgments

In Figure 3, we display results for a set of graded properties annotated in one or more of the datasets we consider. The properties are defined as follows:

- *Acceptability* refers to whether it is plausible or not that a response belongs to a telephonic dialogue and was annotated in Swithboard and Dailydialog;
- *Coherence* was annotated for summaries and model-generated reasonings as part of the datasets NewsRoom, ROSCOE, and SummEval;
- *Consistency* refers to the alignment between facts described in a summary and in its source text, and was annotated in SummEval;
- *Engaging* indicates whether a response generated in the context of a dialogue is dull or interesting and was annotated in TopicalChat and PersonaChat;
- *Fluency* measures whether a piece of text is grammatically correct and well-formatted, and was annotated in NewsRoom, SummEval and Recipe-generation;
- *Informativeness* refers to the extent to which a summary captures the key points of the full text, and was annotated for summaries as part of the NewsRoom dataset;
- *Relevance* refers to whether a summary selects important information as opposed to including redundancies, and was annotated for NewsRoom and SummEval;
- *Verbosity* indicates whether a generated recipe is concise and avoids unnecessary repetitions, and was annotated in Recipe-generation.

E Inference Details

All open-model checkpoints were obtained using the HuggingFace pipeline and we access all proprietary models using their corresponding API libraries. The proprietary models were accessed from 06-06-2024 to 13-06-2024, for standard prompting and from 09-10-2024 to 13-12-2024, for CoT prompting. We obtain the model responses using greedy decoding, which we operationalise for the proprietary models by setting the temperature parameter to 0. We allow open models to generate a maximum of 25 new tokens and proprietary models to generate a maximum of 5 new tokens. For CoT prompting, we allow for a maximum of 1000 new tokens.

We leverage Nvidia A100 (80 GB) GPUs for a total of 321 compute hours. The cost of running experiments using Gemini-1.5-flash was €30.31, while the cost of experiments using GPT-4o was approximately $565.

F Valid Response Rates

Table 5 reports the rate of valid responses for each model and dataset. Valid response rates are summarised per model and dataset in Figures 5 and 6.

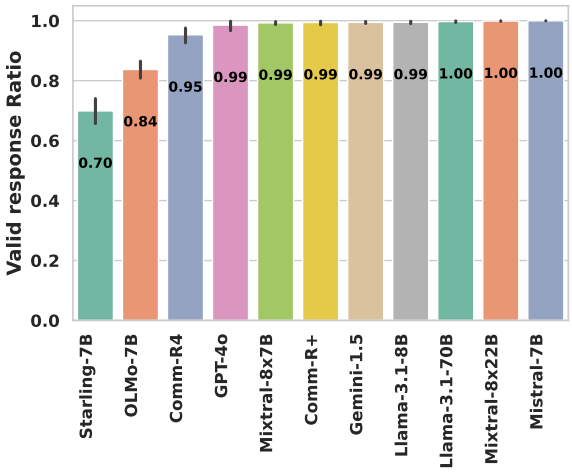


Figure 5: Valid response rate per model.

G More Details on Toxicity and Safety Evaluation

For the Medical-safety dataset, models often refused to answer. Instead they tended to generate explanations, copy what they had in the prompt, or tried to be generally helpful because they saw that it was a medical issue. Since we take a random answer when no answer could be detected, this contributes to lower the results obtained on this task. Scores for the DICES dataset were also low, even though the valid response rate was high, because in this case there is the 'Unsure' option, which (along with 'Unsafe') models preferred over calling anything 'Safe'. For ToxicChat, models performed reasonably well.

H Additional Results

In Table 6 we report human-model alignment scores per dataset for all models tested, thus complementing Table 1 in the paper.

Chain-of-Thought Prompts. For the results with CoT prompting, we use the same original instructions used to gather human judgments as prompts for the model but adapt the additional guidelines to emphasise multi-step reasoning rather than constrain the models' output. Specifically, we append the original instructions with the following additional guideline: '*Always end your answer with either {} regarding the entire context. Let's think step by step.*', in which *{}* is replaced with an enumeration of all possible answer labels formatted as '*Therefore, {label A} is correct, or therefore, {label B} is correct, or therefore [...].*'. This also allows for automatically extracting the final answers from model responses during evaluation. In this study, we evaluate nine models and exclude Mixtral-8x22B and Comm-R+ due to computational constraints. For the CoLa-grammar dataset, we obtain GPT-4o responses only for ten percent of its instances (that are randomly sampled) to address the slow processing times and rate limitations. While CoT prompting leads to improved agreement scores and correlations when used with some models for certain datasets (see Table 7), its overall effectiveness compared to the results obtained using standard prompts without CoT (see Table 6) is inconsistent.

Prompt Paraphrases. We experiment with paraphrased prompts for three datasets that models struggle with: DICES-350-expert, WMT 2023 En-De, and WMT 2023 Zh-En. The paraphrase for dices-350-expert elaborates on the concept of safety, compared to its short original prompt, whereas the paraphrases for the WMT datasets are more concise regarding what comprises a good translation compared to the original. We do not observe consistent improvements when using paraphrased prompts compared to the original prompts (Table 4).

Few-shot Prompts. For the three datasets above—DICES-350-expert, WMT 2023 En-De, and WMT 2023 Zh-En—we also experiment with few-shot prompts (Table 4), where we provide the model with 6 examples for DICES-350-expert, 3 of safe conversations and 3 of unsafe conversations, and 4 examples for each WMT 2023 dataset, 2 of high-scoring translations and 2 of low-scoring translations. Using few-shot prompts does not improve correlations for dices-350-expert. On the WMT 2023 datasets, we observe higher correlations for Llama 3.1 8B but very moderate or no improvements on the other two models. Given that these improvements are inconsistent across datasets, we did not scale up the experiments to all 20 datasets and 11 models.

	Prompt	Llama 3.1 8B	Llama 3.1 70B	Mixtral-8x7B
DICES-350-expert	Original	0.01	-0.13	-0.11
	CoT	-0.07	-0.26	-0.02
	Few-shot	0.01	-0.22	-0.01
	Paraphrase	-0.13	-0.36	-0.09
WMT 2023 En-De	Original	0.08 (1)	0.14 (1)	0.17 (1)
	CoT	0.18 (1)	0.16 (1)	0.20 (1)
	Few-shot	0.19 (1)	0.20 (1)	0.20 (1)
	Paraphrase	0.01 ±0.09 (3)	0.08 ± 0.12 (3)	0.14 ±0.05 (3)
WMT 2023 Zh-En	Original	0.02 (1)	0.14 (1)	0.15 (1)
	CoT	0.13 (1)	0.13 (1)	0.16 (1)
	Few-shot	0.15 (1)	0.14 (1)	0.16 (1)
	Paraphrase	0.08 ±0.04 (3)	0.09 ±0.07 (2)	0.13 ±0.03 (3)

Table 4: Cohen's kappa for DICES-350-expert and Spearman's correlation for two WMT 2023 datasets, comparing the original prompt and CoT prompt to few-shot prompts and prompt paraphrases for a selection of models. For datasets with more than one paraphrased prompt, we report the average and standard deviation across paraphrases. For Spearman's correlations, we report the number of significant correlations ($p < 0.05$) for each model and dataset in brackets.

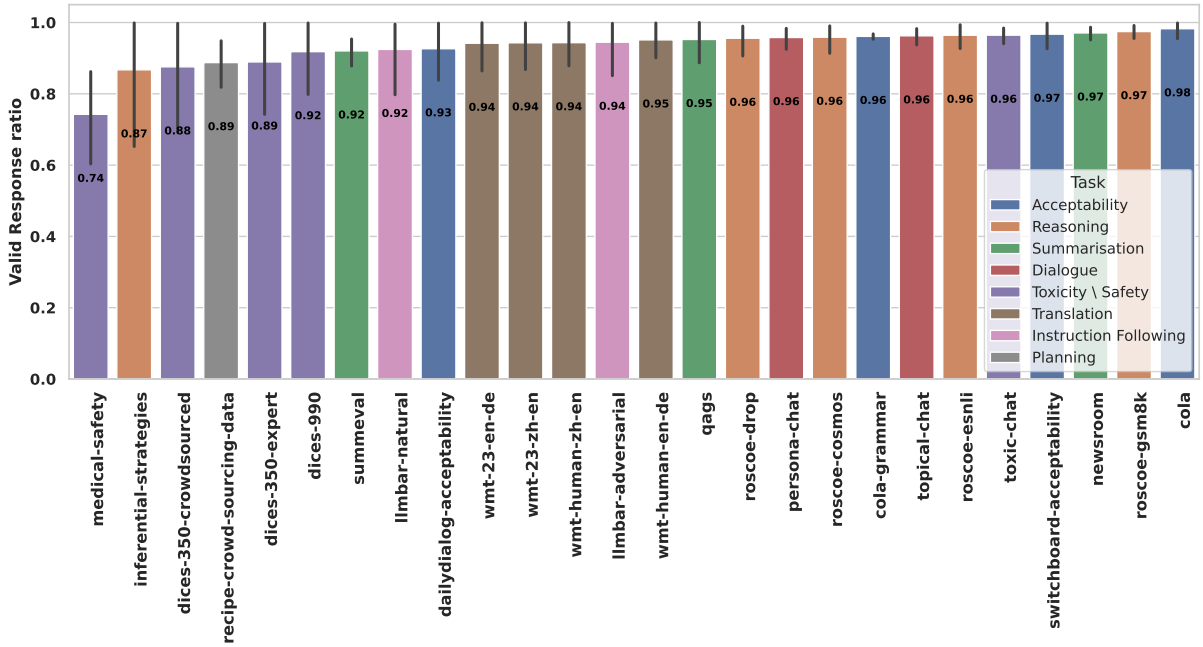


Figure 6: Average ratios of valid responses across datasets over the 11 models we tested.

Type	Dataset (#Subtasks)	GPT-4o	Llama-3.1-70B	Mixtral-8x22B	Gemini-1.5	Mixtral-8x7B	Comm-R+	Comm-R4	Llama-3.1-8B	Mistral-7B	Starling-7B	OLMo-7B
Categorical Annotations	CoLa (1)	1.0	1.0	1.0	1.0	0.98	1.0	1.0	1.0	1.0	0.85	0.98
	CoLa-grammar (63)	1.0	1.0	1.0	1.0	1.0	1.0	1.0±0.01	1.0	1.0	0.71±0.15	0.87±0.11
	LLMBar-natural (1)	1.0	1.0	1.0	1.0	0.95	1.0	0.95	1.0	1.0	0.33	0.94
	LLMBar-adversarial (1)	1.0	1.0	1.0	1.0	0.97	1.0	0.96	1.0	1.0	0.48	0.98
	ToxicChat (2)	1.0	1.0	1.0	0.99	0.96±0.06	0.99	0.91±0.11	0.98	1.0	0.86±0.02	0.92±0.08
	Persona Chat (2)	1.0	1.0	1.0	1.0	0.98±0.02	1.0	0.89±0.15	1.0	1.0	0.96±0.01	0.58±0.24
	Topical Chat (2)	1.0	1.0	1.0	1.0	1.0	1.0	0.99±0.01	1.0	1.0	0.7±0.12	0.77±0.24
	ROSCOE-GSM8K (2)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.92±0.01	0.8±0.23
	ROSCOE-eSNLI (2)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.6±0.33	0.73±0.18
	DICES-990 (1)	1.0	1.0	1.0	0.99	0.98	1.0	1.0	1.0	1.0	0.77	0.35
	Inferential strategies (1)	1.0	1.0	1.0	1.0	0.99	0.97	1.0	1.0	1.0	0.05	0.53
	ROSCOE-CosmosQA (2)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.49±0.45	0.75±0.19
	QAGS (1)	1.0	1.0	1.0	0.97	1.0	1.0	1.0	1.0	1.0	0.73	0.78
	Medical-safety (2)	0.35±0.37	0.96±0.02	0.97±0.04	0.97±0.04	0.85±0.1	0.78±0.31	0.33±0.47	0.89±0.11	1.0	0.22±0.08	0.85±0.19
	DICES-350-expert (1)	1.0	1.0	1.0	0.99	1.0	0.98	0.99	1.0	1.0	0.55	0.27
	DICES-350-crowdsourced (1)	1.0	1.0	1.0	0.99	0.99	0.98	1.0	1.0	1.0	0.51	0.16
	ROSCOE-DROP (2)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.51±0.51	0.68±0.26
Graded Annotations	Dailydialog (1)	1.0	1.0	1.0	0.99	1.0	1.0	0.69	1.0	1.0	0.89	0.62
	Switchboard (1)	1.0	1.0	1.0	1.0	0.99	1.0	0.93	1.0	1.0	0.95	0.77
	Persona Chat (4)	1.0	1.0	1.0	1.0	1.0	1.0	0.97±0.03	1.0	1.0	0.71±0.27	0.92±0.15
	Topical Chat (4)	1.0	1.0	1.0	1.0	1.0	1.0	0.99±0.01	1.0	1.0	0.75±0.1	0.91±0.07
	Recipe-generation (6)	1.0	1.0	1.0	1.0	1.0±0.01	1.0	0.67±0.2	1.0	1.0	0.11±0.16	0.98±0.01
	ROSCOE-CosmosQA (2)	1.0	1.0	1.0	1.0	1.0	1.0	0.99±0.01	1.0	1.0	0.97±0.01	0.89
	ROSCOE-DROP (2)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.99±0.01	0.85±0.11
	ROSCOE-eSNLI (2)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.89
	ROSCOE-GSM8K (2)	1.0	1.0	1.0	1.0	1.0	1.0	0.98	1.0	1.0	0.84±0.02	0.91±0.06
	NewsRoom (4)	1.0	1.0	0.98±0.01	0.99	1.0	1.0	1.0	1.0	1.0	0.89±0.1	0.83±0.04
	SummEval (4)	0.87±0.13	0.94±0.06	1.0	0.9±0.06	1.0	1.0	0.72±0.3	0.94±0.08	1.0	0.96±0.04	0.79±0.05
	WMT 2020 En-De (1)	1.0	1.0	1.0	0.99	0.87	1.0	1.0	1.0	1.0	0.85	0.75
	WMT 2020 Zh-En (1)	1.0	1.0	1.0	1.0	0.87	1.0	1.0	1.0	1.0	0.81	0.7
	WMT 2023 En-De (1)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.79	0.58
	WMT 2023 Zh-En (1)	1.0	1.0	1.0	1.0	0.99	1.0	1.0	1.0	1.0	0.78	0.61

Table 5: Ratios of valid responses per dataset for all models we evaluate.

Type	Dataset (# properties judged)	GPT-4o	Llama-3.1-70B	Mixtral-8x22B	Gemini-1.5	Mixtral-8x7B	Comm-R+	Comm-R4	Llama-3.1-8B	Mistral-7B	Starling-7B	OLMo-7B
Categorical Annotations	CoLa (1)	0.34	0.46	0.54	0.45	0.55	0.12	0.01	0.42	0.43	0.45	0.42
	CoLa-grammar (63)	0.47 ±0.22	0.28 ±0.24	0.28 ±0.23	0.26 ±0.24	0.21 ±0.18	0.13 ±0.14	0.08 ±0.1	0.1 ±0.14	0.09 ±0.13	0.07 ±0.08	0.04 ±0.06
	LLMBar-natural (1)	0.84	0.8	0.72	0.79	0.54	0.56	0.59	0.57	0.3	0.28	0.24
	LLMBar-adversarial (1)	0.58	0.46	0.2	0.29	0.06	0.11	-0.2	-0.18	-0.2	-0.12	-0.1
	ToxicChat (2)	0.49 ±0.36	0.41 ±0.26	0.45 ±0.27	0.45 ±0.35	0.36 ±0.12	0.28 ±0.35	0.2 ±0.21	0.34 ±0.29	0.45 ±0.18	0.27 ±0.26	0.3 ±0.13
	Persona Chat (2)	0.24 ±0.34	0.24 ±0.33	0.58 ±0.59	-0.03 ±0.04	0.54 ±0.65	0.48 ±0.74	0.01 ±0.01	0.5 ±0.7	0.47 ±0.75	-0.03 ±0.04	0.02 ±0.03
	Topical Chat (2)	0.05 ±0.07	-0.02 ±0.02	-0.03 ±0.04	-0.03 ±0.04	0.02 ±0.03	0.01 ±0.02	0.01 ±0.01	0.57 ±0.61	-0.03 ±0.05	0.04 ±0.06	0.03 ±0.04
	ROSCOE-GSM8K (2)	0.59 ±0.35	0.64 ±0.27	0.62 ±0.38	0.6 ±0.24	0.58 ±0.36	0.0	0.21 ±0.03	0.36 ±0.31	0.47 ±0.34	-0.03 ±0.01	-0.01 ±0.02
	ROSCOE-eSNLI (2)	0.29 ±0.06	0.38 ±0.08	0.13 ±0.13	0.11 ±0.18	0.1 ±0.11	0.03 ±0.05	-0.01 ±0.01	0.14 ±0.2	0.02 ±0.09	0.01 ±0.07	-0.04 ±0.09
	ROSCOE-DROP (2)	0.29 ±0.08	0.27 ±0.07	0.2 ±0.12	0.08 ±0.05	0.13 ±0.21	0.03 ±0.04	0.02 ±0.07	0.02 ±0.02	0.09 ±0.08	0.01 ±0.03	0.0 ±0.01
	ROSCOE-CosmosQA (2)	0.16 ±0.07	0.25 ±0.02	0.09 ±0.17	0.14 ±0.17	0.19 ±0.05	-0.03 ±0.01	-0.01 ±0.02	0.08 ±0.11	0.29 ±0.03	0.03	-0.18
	QAGS (1)	0.72	0.7	0.66	0.65	0.68	0.13	0.33	0.58	0.43	0.02	0.11
	Medical-safety (2)	0.01 ±0.03	-0.03 ±0.06	-0.02 ±0.09	-0.03 ±0.08	0.0 ±0.06	0.01 ±0.02	0.01 ±0.01	0.01	-0.03 ±0.12	0.0 ±0.02	-0.02 ±0.07
	DICES-990 (1)	-0.24	-0.17	-0.16	-0.12	-0.2	-0.09	-0.02	-0.11	-0.12	-0.05	0.0
	DICES-350-expert (1)	-0.2	-0.13	-0.15	-0.03	-0.11	0.01	0.01	0.01	0.01	0.01	-0.06
	DICES-350-crowdsourced (1)	-0.22	-0.18	-0.08	-0.02	-0.11	-0.08	0.01	-0.05	-0.04	0.01	-0.03
	Inferential strategies (1)	0.42	0.4	0.02	0.22	0.06	-0.02	-0.12	0.13	0.01	0.01	0.04
Graded Annotations	Dailydialog (1)	0.69 (1)	0.6 (1)	0.55 (1)	0.63 (1)	0.63 (1)	0.52 (1)	0.23 (1)	0.61 (1)	0.48 (1)	0.09 (0)	0.07 (0)
	Switchboard (1)	0.66 (1)	0.45 (1)	0.63 (1)	0.59 (1)	0.56 (1)	0.36 (1)	0.53 (1)	0.28 (1)	0.52 (1)	0.13 (0)	0.3 (1)
	Persona Chat (4)	0.22 ±0.11 (2)	-0.02 ±0.2 (0)	0.16 ±0.1 (1)	0.1 ±0.09 (0)	0.02 ±0.15 (0)	0.07 ±0.13 (0)	0.05 ±0.2 (0)	-0.02 ±0.14 (0)	-0.09 ±0.17 (1)	0.03 ±0.13 (0)	-0.06 ±0.14 (0)
	Topical Chat (4)	0.26 ±0.03 (2)	0.28 ±0.1 (2)	0.13 ±0.04 (0)	0.17 ±0.12 (1)	0.21 ±0.18 (1)	0.14 ±0.05 (0)	0.07 ±0.07 (0)	0.15 ±0.13 (0)	0.29 ±0.11 (3)	0.14 ±0.16 (1)	0.08 ±0.21 (1)
	Recipe-generation (6)	0.78 ±0.05 (6)	0.66 ±0.07 (6)	0.6 ±0.15 (6)	0.67 ±0.09 (5)	0.57 ±0.24 (5)	0.32 ±0.28 (5)	0.06 ±0.26 (3)	0.34 ±0.09 (5)	0.28 ±0.08 (4)	0.04 ±0.17 (1)	0.1 ±0.08 (0)
	ROSCOE-GSM8K (2)	0.82 ±0.12 (2)	0.83 ±0.11 (2)	0.81 ±0.14 (2)	0.81 ±0.12 (2)	0.79 ±0.13 (2)	0.68 ±0.2 (2)	0.7 ±0.08 (2)	0.76 ±0.15 (2)	0.63 ±0.18 (2)	0.46 ±0.13 (2)	0.1 ±0.07 (1)
	ROSCOE-eSNLI (2)	0.49 ±0.24 (2)	0.4 ±0.16 (2)	0.38 ±0.17 (2)	0.35 ±0.21 (2)	0.32 ±0.12 (2)	0.09 ±0.08 (0)	0.28 ±0.21 (1)	0.19 ±0.16 (1)	0.32 ±0.12 (2)	0.11 ±0.06 (0)	0.11 ±0.17 (1)
	ROSCOE-DROP (2)	0.57 ±0.22 (2)	0.59 ±0.16 (2)	0.44 ±0.15 (2)	0.44 ±0.13 (2)	0.32 ±0.12 (2)	0.21 ±0.22 (1)	0.37 ±0.18 (2)	0.23 ±0.1 (2)	0.22 ±0.22 (1)	0.16 ±0.17 (1)	0.15 ±0.21 (1)
	ROSCOE-CosmosQA (2)	0.57 ±0.18 (2)	0.55 ±0.18 (2)	0.51 ±0.16 (2)	0.57 ±0.17 (2)	0.53 ±0.21 (2)	0.33 ±0.25 (2)	0.48 ±0.17 (2)	0.44 ±0.26 (2)	0.57 ±0.2 (2)	0.13 ±0.04 (1)	0.49 ±0.24 (2)
	NewsRoom (4)	0.59 ±0.02 (4)	0.59 ±0.03 (4)	0.44 ±0.05 (4)	0.55 ±0.03 (4)	0.5 ±0.07 (4)	0.36 ±0.06 (4)	0.16 ±0.05 (4)	0.45 ±0.04 (4)	0.26 ±0.06 (4)	0.21 ±0.08 (4)	-0.01 ±0.04 (0)
	SummEval (4)	0.35 ±0.06 (4)	0.44 ±0.14 (4)	0.54 ±0.08 (4)	0.38 ±0.02 (4)	0.48 ±0.02 (4)	0.19 ±0.06 (4)	0.13 ±0.06 (4)	0.29 ±0.09 (4)	0.4 ±0.12 (4)	0.15 ±0.05 (4)	0.06 ±0.02 (2)
	WMT 2020 En-De (1)	0.63 (1)	0.37 (1)	0.51 (1)	0.46 (1)	0.2 (1)	0.42 (1)	0.15 (1)	0.11 (1)	0.36 (1)	0.15 (1)	-0.03 (1)
	WMT 2020 Zh-En (1)	0.54 (1)	0.39 (1)	0.48 (1)	0.41 (1)	0.25 (1)	0.42 (1)	0.15 (1)	0.14 (1)	0.39 (1)	0.15 (1)	0.01 (0)
	WMT 2023 En-De (1)	0.22 (1)	0.14 (1)	0.23 (1)	0.16 (1)	0.17 (1)	0.22 (1)	0.19 (1)	0.08 (1)	0.18 (1)	-0.09 (1)	-0.05 (1)
	WMT 2023 Zh-En (1)	0.17 (1)	0.14 (1)	0.19 (1)	0.14 (1)	0.15 (1)	0.15 (1)	0.14 (1)	0.02 (1)	0.15 (1)	0.01 (0)	0.01 (0)

Table 6: Scores per dataset for all models we evaluate: Cohen's kappa for categorical annotations and Spearman's correlation for graded annotations. For Spearman's correlations, we report the number of significant correlations ($p < 0.05$) for each model and dataset in brackets. Datasets in blue concern human-generated language while those in red concern model-generated text.

Type	Dataset (#properties judged)	GPT-4o	Llama-3.1-70B	Gemini-1.5	Mixtral-8x7B	Comm-R4	Llama-3.1-8B	Mistral-7B	Starling-7B	OLMo-7B
Categorical Annotations	CoLa (1)	0.35	0.41	0.45	0.47	0.3	0.35	0.51	0.39	0.26
	CoLa-grammar (63)	-0.04 ±0.06	0.35 ±0.25	0.33 ±0.23	0.21 ±0.16	0.05 ±0.09	0.24 ±0.21	0.19 ±0.19	0.16 ±0.16	0.04 ±0.06
	LLMBar-natural (1)	0.86	0.86	0.71	0.62	0.37	0.55	0.56	0.46	0.21
	LLMBar-adversarial (1)	0.67	0.92	0.32	-0.07	-0.25	-0.3	-0.29	-0.25	-0.05
	ToxicChat (2)	0.42 ±0.1	0.37 ±0.03	0.41 ±0.36	0.33 ±0.21	0.33 ±0.26	0.22 ±0.03	0.41 ±0.07	0.33 ±0.16	0.31 ±0.2
	Persona Chat (2)	0.83 ±0.25	0.13 ±0.19	0.57 ±0.6	0.0 ±0.01	0.47 ±0.75	-0.01 ±0.01	-0.01 ±0.02	-0.03 ±0.05	-0.03 ±0.05
	Topical Chat (2)	0.57 ±0.61	0.09 ±0.13	0.03 ±0.04	-0.02 ±0.03	0.48 ±0.74	-0.0	-0.03 ±0.05	-0.03 ±0.05	-0.03 ±0.05
	ROSCOE-GSM8K (2)	0.29 ±0.77	0.52 ±0.26	0.52 ±0.25	-0.29 ±0.02	-0.24 ±0.34	0.12 ±0.15	0.38 ±0.46	-0.06 ±0.18	-0.04 ±0.03
	ROSCOE-eSNLI (2)	0.05 ±0.16	0.1 ±0.09	-0.01 ±0.01	-0.03 ±0.05	-0.04 ±0.04	-0.04 ±0.04	-0.01 ±0.09	0.06 ±0.17	-0.03 ±0.06
	ROSCOE-DROP (2)	-0.05 ±0.01	0.13 ±0.15	-0.08 ±0.07	-0.11 ±0.15	-0.14 ±0.12	-0.05 ±0.05	-0.07 ±0.09	-0.03	-0.07 ±0.04
	ROSCOE-CosmosQA (2)	-0.29 ±0.06	0.03	-0.26 ±0.09	-0.29 ±0.12	-0.3 ±0.05	-0.11 ±0.16	-0.25 ±0.2	-0.09 ±0.16	-0.23 ±0.12
	QAGS (1)	0.69	0.7	0.66	0.66	0.34	0.58	0.46	0.49	0.07
	Medical-safety (2)	-0.01 ±0.09	-0.02 ±0.08	-0.01 ±0.09	0.03 ±0.07	-0.0 ±0.01	-0.02 ±0.01	-0.02 ±0.07	-0.01 ±0.01	-0.01 ±0.09
	DICES-990 (1)	-0.22	-0.15	-0.16	-0.14	-0.1	-0.16	-0.08	-0.08	-0.12
	DICES-350-expert (1)	-0.3	-0.26	-0.06	-0.02	-0.13	-0.07	0.06	0.01	-0.04
	DICES-350-crowdsourced (1)	-0.26	-0.21	-0.13	-0.02	-0.18	-0.19	0.0	-0.07	0.06
	Inferential strategies (1)	0.47	0.4	0.25	0.13	0.01	0.04	-0.01	0.12	0.02
Graded Annotations	Dailydialog (1)	0.69 (1)	0.62 (1)	0.56 (1)	0.25 (1)	0.42 (1)	0.51 (1)	0.4 (1)	0.34 (1)	0.27 (1)
	Switchboard (1)	0.6 (1)	0.48 (1)	0.53 (1)	0.07 (0)	0.38 (1)	0.17 (0)	0.36 (1)	0.35 (1)	0.08 (0)
	Persona Chat (4)	0.2 ±0.09 (2)	0.09 ±0.2 (1)	0.11 ±0.06 (0)	0.06 ±0.15 (0)	0.17 ±0.21 (1)	-0.04 ±0.22 (1)	0.04 ±0.13 (0)	-0.01 ±0.22 (1)	0.06 ±0.18 (0)
	Topical Chat (4)	0.22 ±0.02 (0)	0.14 ±0.13 (1)	0.11 ±0.1 (0)	0.06 ±0.18 (1)	0.1 ±0.14 (0)	0.16 ±0.17 (1)	0.25 ±0.05 (2)	0.17 ±0.09 (1)	0.08 ±0.13 (0)
	Recipe-generation (6)	0.67 ±0.12 (6)	0.64 ±0.14 (6)	0.65 ±0.09 (6)	0.42 ±0.18 (5)	0.14 ±0.15 (2)	0.31 ±0.15 (2)	0.41 ±0.07 (6)	0.34 ±0.2 (4)	0.09 ±0.1 (0)
	ROSCOE-GSM8K (2)	0.82 ±0.12 (2)	0.81 ±0.12 (2)	0.81 ±0.11 (2)	0.81 ±0.13 (2)	0.49 ±0.13 (2)	0.8 ±0.11 (2)	0.58 ±0.13 (2)	0.64 ±0.15 (2)	0.07 ±0.07 (0)
	ROSCOE-eSNLI (2)	0.49 ±0.29 (2)	0.39 ±0.38 (1)	0.31 ±0.32 (1)	0.31 ±0.09 (2)	0.33 ±0.01 (2)	0.23 ±0.17 (1)	0.17 ±0.03 (1)	0.19 ±0.13 (1)	-0.05 ±0.07 (0)
	ROSCOE-DROP (2)	0.54 ±0.17 (2)	0.55 ±0.19 (2)	0.44 ±0.12 (2)	0.29 ±0.14 (2)	0.29 ±0.24 (1)	0.44 ±0.07 (2)	0.28 ±0.03 (2)	0.17 ±0.15 (1)	-0.04 ±0.02 (0)
	ROSCOE-CosmosQA (2)	0.57 ±0.21 (2)	0.55 ±0.22 (2)	0.56 ±0.11 (2)	0.55 ±0.12 (2)	0.62 ±0.15 (2)	0.38 ±0.06 (2)	0.54 ±0.21 (2)	0.32 ±0.14 (2)	0.3 ±0.14 (1)
	NewsRoom (4)	0.57 ±0.05 (4)	0.53 ±0.03 (4)	0.53 ±0.05 (4)	0.46 ±0.02 (4)	0.19 ±0.06 (4)	0.49 ±0.04 (4)	0.19 ±0.04 (4)	0.22 ±0.12 (3)	0.09 ±0.06 (1)
	SummEval (4)	0.45 ±0.11 (4)	0.48 ±0.16 (4)	0.33 ±0.03 (4)	0.35 ±0.05 (4)	0.17 ±0.05 (4)	0.24 ±0.13 (4)	0.38 ±0.1 (4)	0.24 ±0.11 (4)	0.03 ±0.07 (1)
	WMT 2020 En-De (1)	0.57	0.44 (1)	0.38 (1)	0.39 (1)	0.13 (1)	0.34 (1)	0.33 (1)	0.3 (1)	0.0 (0)
	WMT 2020 Zh-En (1)	0.52 (1)	0.44 (1)	0.44 (1)	0.42 (1)	0.19 (1)	0.36 (1)	0.39 (1)	0.35 (1)	0.13 (1)
	WMT 2023 En-De (1)	0.2 (1)	0.16 (1)	0.18 (1)	0.2 (1)	0.2 (1)	0.18 (1)	0.21 (1)	0.19 (1)	-0.01 (0)
	WMT 2023 Zh-En (1)	0.15 (1)	0.13 (1)	0.15 (1)	0.16 (1)	0.16 (1)	0.13 (1)	0.11 (1)	0.13 (1)	0.06 (0)

Table 7: Scores per dataset for all models we evaluate using CoT prompts: Cohen's kappa for categorical annotations and Spearman's correlation for graded annotations. For Spearman's correlation, we report the number of significant correlations for each model and dataset in brackets. Datasets in blue concern human-generated language while those in red concern model-generated text.

FocalPO: Enhancing Preference Optimizing by Focusing on Correct Preference Rankings

Tong Liu[1,5], **Xiao Yu**[2†], **Wenxuan Zhou**[3], **Jindong Gu**[4], **Volker Tresp**[1,5]
[1]LMU Munich [2]Columbia University [3]University of Southern California
[4]University of Oxford [5]Munich Center for Machine Learning
tongliu.physics@gmail.com xy2437@columbia.edu

Abstract

Efficient preference optimization algorithms such as Direct Preference Optimization (DPO) have become a popular approach in aligning large language models (LLMs) with human preferences. These algorithms implicitly treat the LLM as a reward model, and focus on training it to correct misranked preference pairs. However, recent work (Chen et al., 2024) empirically finds that DPO training *rarely improves these misranked preference pairs*, despite its gradient emphasizing on these cases. We introduce FocalPO, a DPO variant that instead *down-weighs* misranked preference pairs and prioritizes enhancing the model's understanding of pairs that it can already rank correctly. Inspired by Focal Loss used in vision tasks, FocalPO achieves this by adding a modulating factor to dynamically scale DPO loss. Our experiment demonstrates that FocalPO surpasses DPO and its variants on popular benchmarks like Alpaca Eval 2.0 and Arena-Hard using Mistral-Base-7B and Llama-3-Instruct-8B, with the introduced hyperparameter fixed. Additionally, we empirically reveals how FocalPO affects training on correct and incorrect sample groups, further underscoring its effectiveness [1].

1 Introduction

Reinforcement learning from human feedback (RLHF) played a crucial role in aligning large language models (LLMs) with human preferences (Christiano et al., 2017; Ouyang et al., 2022; Stiennon et al., 2020). However, conducting RLHF with Proximal Policy Optimization (PPO, Schulman et al. 2017) is computationally expensive. Therefore, recent works have studied more efficient approaches, such as Direct Preference Optimization (DPO, Rafailov et al. 2023) and its variants (Amini et al., 2024; Meng et al., 2024; Ethayarajh et al., 2024; Zhou et al., 2024), which implicitly treat the language model itself as a reward model and directly optimizes it using preference datasets.

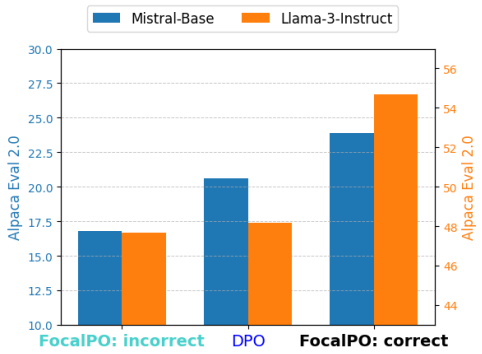


Figure 1: Comparing the effectiveness of up-weighing **incorrect** preference pairs against up-weighing **correct** preference pairs with FocalPO during training. We find that performance further improves by prioritizing learning from pairs that can already be ranked correctly.

Despite its popularity, recent work (Chen et al., 2024) shows that DPO often *fails to correct incorrect preference rankings made by the implicit reward model prior to training*, despite its gradient emphasizing these cases. While one recent competitive DPO variant SimPO (Meng et al., 2024) introduces a fixed margin term in the preference optimization loss, we find it *inherently reduces* the disproportionately high weights assigned to learning incorrect response pairs. ODPO (Amini et al., 2024) further incorporates additional labeled reward values and employs an instance-dependent margin term. However, while these approaches implicitly re-weigh instances through the use of margin terms, they lack explicit and dynamic mechanisms for such re-weighing.

To this end, we propose FocalPO, a loss function that *dynamically and explicitly reduces* DPO's emphasis on learning response pairs that the model struggles to rank while prioritizing preference pairs with accurate implicit reward estimates. Inspired by *Focal Loss* (Lin et al., 2020) in vision tasks, we

[1]Code available at: https://github.com/TongLiu-github/focalpo

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 256–267
July 27 - August 1, 2025

achieve this by adding a modulating factor of p^γ to dynamically scale the DPO loss, $-\log p$. Here, p represents the probability that the preferred response over the dispreferred one. Intuitively, such a factor automatically up-weighs the contribution of correctly ranked samples during training, with the scaling factor decays to zero as preference probability increases. Gradient analysis shows that FocalPO equivalently assigns higher weights to pairs whose implicit reward estimate order is accurate and lower weights to incorrect pairs. This encourages models to learn from preference pairs with correct reward estimates while avoiding excessive influence from challenging ones. Experiments show that FocalPO significantly outperforms the original DPO. Additionally, we provide case study showing how FocalPO affects training on correct and incorrect sample groups, further underscoring its effectiveness.

Our contributions are summarized as follows: (1) We introduce FocalPO, a novel loss function based on DPO and Focal Loss, which emphasizes learning preference pairs with accurate implicit reward estimates. (2) We show that FocalPO surpasses DPO as well as many DPO variants on popular chat benchmarks like Alpaca-Eval 2.0 and Arena-Hard, using both Mistral-Base-7B and Llama-3-Instruct-8B models. (3) We empirically demonstrate the impact of prioritizing learning correctly vs. incorrect ranked preference pairs.

2 Background

Given an instruction-tuned language model π_θ, Direct Preference Optimization (DPO, Rafailov et al. (2023)) further optimizes it as an implicit reward model using a preference dataset. Let $\mathcal{D} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_i^N$ denote a preference dataset where $x^{(i)} \in \mathcal{X}$ is a prompt and $(y_w^{(i)}, y_l^{(i)})$ is a pair of answers, with the preference of $y_w^{(i)} \succ y_l^{(i)}$ expressed by human or AI annotators. Under the assumption of Bradley-Terry model (Bradley and Terry, 1952), the human preference distribution p^* can be approximated as:

$$p^*(y_w^{(i)} \succ y_l^{(i)} \mid x^{(i)}) = \sigma(r^*(x^{(i)}, y_w^{(i)}) - r^*(x^{(i)}, y_l^{(i)})), \tag{1}$$

where σ and $r^*(x, y)$ are the sigmoid function and the latent reward model used to generate the ground-truth preference, respectively. To model this human preference, DPO uses a reparametrization trick to express it in terms of the optimal policy π^*: $r^*(x, y) = \beta \log \frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x)$, where $Z(x)$ is the partition function only based on x. Applying the above equations, the maximum likelihood estimation objective of DPO is:

$$\begin{aligned}\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) &= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log p\left(y_w \succ y_l \mid x\right)\right] \\ &= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)}\right.\right. \\ &\quad \left.\left. -\beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right],\end{aligned} \tag{2}$$

which *emphasizes* on learning responses pairs that the model struggle to rank (i.e., wrong reward estimates). However, recent work such as Chen et al. (2024) shows that DPO cannot learn to correct these response pairs, and a few other competitive variants such as SimPO (Meng et al., 2024) and ODPO (Amini et al., 2024) *implicitly introduce a margin term* that reduces such emphasis.

3 FocalPO: Focal loss inspired Preference Optimization

3.1 FocalPO loss

We propose FocalPO loss, which directly scales DPO's loss to reduce emphasis on learning response pairs the model struggles to rank and focuses on preference pairs with accurate implicit reward estimates. Inspired by Focal Loss (Lin et al., 2020) in vision tasks (i.e., adding a modulating factor $(1-p)^\gamma$ to the cross entropy loss), we replace the original cross-entropy loss in Eq. 2 with:

$$\begin{aligned}&\mathcal{L}_{\text{FocalPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \\ &\left[\left(1 - p\left(y_w \succ y_l \mid x\right)\right)^{-\gamma} \cdot \log p\left(y_w \succ y_l \mid x\right)\right]\end{aligned} \tag{3}$$

$$\begin{aligned}&\approx -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[p\left(y_w \succ y_l \mid x\right)^\gamma \log p\left(y_w \succ y_l \mid x\right)\right] \\ &\quad + \text{constant},\end{aligned} \tag{4}$$

where $\gamma \geq 0$ is a tunable *focusing* hyperparameter. Here we take negative index $-\gamma$ to replace the original positive term γ in Focal Loss, in order to focus on the easier or correctly ranked preference pairs. To simplify the optimization process, we further approximate $(1-p)^{-\gamma} \log p$ [2] with the term $p^\gamma \log p$. Unless otherwise specified, we use Eq. 4 as FocalPO for the rest of the paper. Figure 2 illustrates the focal loss with modulating factors $(1-p)^\gamma$ and p^γ. Converse to $(1-p)^\gamma$,

[2] In experiments, we also observe that training with the power of $-\gamma$ often causes the gradient to become NaN. We reuse the symbol γ for notational simplicity, although the exponent arises from an approximation.

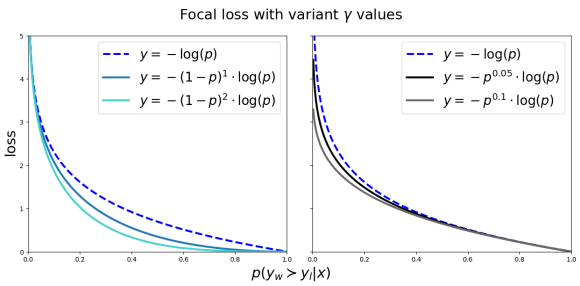


Figure 2: We scale DPO loss using *Focal Loss*. Unlike the original Focal Loss (left), we add a scaling factor of p^γ (right) to the dynamically adjust the cross entropy loss. This modification enables a sample-wise adjustment of weights, reducing the influence of high weights when the implicit reward model order is incorrect.

a higher γ in p^γ leads to a less emphasis placed on hard negative samples. When the implicit reward model "correctly classifies" a preference pair (i.e., $p(y_w \succ y_l|x)$ is large), the modulating factor approaches 1, leaving the loss *almost unchanged*. Conversely, as $p(y_w \succ y_l|x) \to 0$, the modulating factor approaches 0, effectively *down-weighing* the preference pair in the DPO loss.

3.2 Gradient analysis

To better understand the impact of modulating factor on the DPO loss, we perform a gradient analysis. The gradient of DPO with respect to θ is:

$$\nabla_\theta \mathcal{L}_{\text{DPO}} = -\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\sigma(\hat{r}_\theta(y_l) - \hat{r}_\theta(y_w)) \cdot \nabla_\theta \frac{\log \pi(y_w|x)}{\log \pi(y_l|x)}\Big], \tag{5}$$

where $\hat{r}_\theta(y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}$ is the reward implicitly defined by the language model π_θ and reference model π_{ref}. The term $\sigma(\hat{r}_\theta(y_l) - \hat{r}_\theta(y_w))$ assigns higher weights when reward estimate is wrong.

In comparison, the gradient of FocalPO [3] is:

$$\begin{aligned}\nabla_\theta \mathcal{L}_{\text{FocalPO}} = -\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \big[[&\sigma(\hat{r}_\theta(y_l) - \hat{r}_\theta(y_w)) \\ &\cdot \sigma^\gamma(\hat{r}_\theta(y_w) - \hat{r}_\theta(y_l)) + \log \sigma(\hat{r}_\theta(y_w) - \hat{r}_\theta(y_l)) \\ &\cdot \gamma\sigma^\gamma(\hat{r}_\theta(y_w) - \hat{r}_\theta(y_l)) \cdot \sigma(\hat{r}_\theta(y_l) - \hat{r}_\theta(y_w))] \\ &\cdot \nabla_\theta \frac{\log \pi(y_w|x)}{\log \pi(y_l|x)}\Big].\end{aligned} \tag{6}$$

We illustrate the influence of each term in Fig. 3. The blue curve represents the gradient term from DPO, while the red and green curves correspond to the additional gradient terms introduced by our

[3]Refer to Appendix B for detailed derivation and Appendix A for an interpretation of FocalPO.

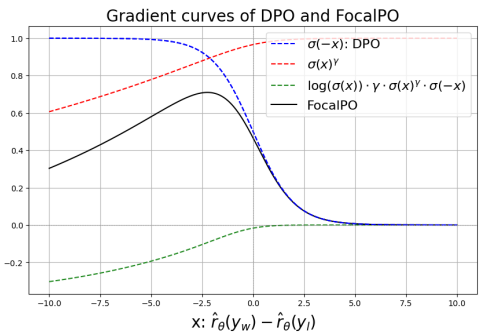


Figure 3: Gradients of DPO and FocalPO. We used $\gamma = 0.05$ for p^γ in FocalPO in this example.

modulating factor. Compared to DPO, the new gradient terms from FocalPO assign lower weights to incorrect preference pairs. When combined, as shown in the black curve in Fig. 3, FocalPO forms *a bell-shaped curve*, assigning the highest weights to preference pairs with reward margins close to zero while assigning lower weights for both correctly classified pairs and significantly incorrect pairs. FocalPO thus aims to mitigate the impact of preference pairs where reward estimates strongly disagree with the preference labels, and shifting focus toward learning preference pairs the model already ranks correctly.

4 Experimental setup

Model and training datasets We perform preference optimization on two representative models, Mistral-Base SFT (7B) [4] and instruction-tuned Llama-3 (8B) [5]. We perform perference learning on the UltraFeedback dataset (Cui et al., 2023) [6] for Mistral-Base, and on the Llama3-ultrafeedback-armorm dataset [7] for Llama-3-Instruct. The former dataset is sampled from multiple LLMs and judged by GPT-4, and the later contains preference pairs generated by Llama-3-Instruct and judged by ArmoRM (Wang et al., 2024).

Hyperparameters For Mistral-Base (7B), we adopt the official hyperparameters from Zephyr (Tunstall et al., 2023) and use $\beta = 0.01$, epoch=1, batch size as 128 and learning rate as 5e-7. For instruction-tuned Llama-3 (8B), we

[4]https://huggingface.co/HuggingFaceH4/mistral-7b-sft-beta

[5]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[6]https://huggingface.co/datasets/trl-lib/ultrafeedback_binarized

[7]https://huggingface.co/datasets/princeton-nlp/llama3-ultrafeedback-armorm

Models	Llama-3-Instruct (8B)			Mistral-Base (7B)		
	Alpaca Eval 2.0		Arena-Hard	Alpaca Eval 2.0		Arena-Hard
	WR	LCWR	WR	WR	LCWR	WR
ORPO	33.8	38.1	26.0	12.6	14.7	6.2
KTO	31.8	33.1	11.7	12.3	14.9	8.8
DPO	47.5	48.2	33.1	18.6	20.6	16.4
SimPO	47.5	53.7	33.8	**21.4**	17.0	13.3
FocalPO	**49.8**	**54.7**	**34.6**	20.4	**23.9**	**17.1**

Table 1: Alpaca Eval 2.0 and Arena-Hard results of DPO, its variants and FocalPO for preference optimazation of Mistral-Base (7B) and Llama-3-Instruct (8B).

follow the setting of SimPO (Meng et al., 2024). We use a batch size of 128, epoch=1, and for each algorithm, we report the best performance after a grid searching learning rates in [3e-7, 5e-7, 6e-7, 1e-6]. We also fix the hyperparameter of γ in FocalPO as 0.05 for both models to prevent possible overfitting of hyperparameters.

Baselines We compare FocalPO against DPO, as well as other popular DPO variants including KTO (Ethayarajh et al., 2024), ORPO (Hong et al., 2024), and SimPO (Meng et al., 2024).

Evaluation To compare the effectiveness of preference learning, we use Alpaca Eval 2.0 (Li et al., 2023; Dubois et al., 2024) and Arena-Hard (Li et al., 2024). Alpaca Eval 2.0 includes 805 representative instructions drawn from multiple datasets and compares model's responses with reference responses generated by `gpt-4-turbo`. We use default `gpt-4-turbo` as the judge model. We present metrics of the win rate (WR) and the length-controlled win rate (LCWR). Arena-Hard contains 500 well-defined challenging prompts and compares model's responses with reference responses generated by `gpt-4-0314`. We use default `gpt-4-1106-preview` as the judge model. We present the metric of WR. Both benchmarks are evaluated under the standard setting.

5 Results

Effectiveness of FocalPO Table 1 demonstrates that FocalPO surpasses DPO and its variants on Alpaca Eval 2.0 and Arena-Hard. Notably, FocalPO improves upon SimPO, a highly performant alignment algorithm, without removing the reference model or additionally adding a sequence length control factor. On Arena-Hard, FocalPO also shows competitive performance compared to other variants, achieving a the best score with Llama-3-Instruct and Mistral-Base-7B. We also present a qualitative analysis comparing outcomes between DPO and FocalPO, detailed in Appendix F. Besides, the analysis of downstream task evaluation is provided in Appendix D.

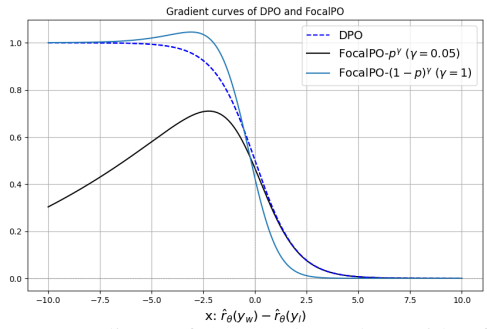


Figure 4: Gradients of DPO and FocalPO with a focus on different sample groups where the reward estimate is correct (p^γ, $\gamma = 0.05$) or incorrect ($(1-p)^\gamma$, $\gamma = 1$).

Focus learning correct/incorrect samples FocalPO down-weighs incorrectly ranked instances that the model struggles to learn, and placing greater emphasis on correct ones compared to DPO. To examine the impact of this choice, we empirically compare two FocalPO settings against DPO: emphasize learning *incorrect* samples using $\gamma = 1$ in the factor $(1-p)^\gamma$; and emphasize learning correct samples using $\gamma = 0.05$ in p^γ. Corresponding gradient curves are illustrated in Fig. 4. For a fair comparison, DPO is evaluated using its official hyperparameter setting while the learning rate is kept consistent across models focusing on incorrect and correct samples. We present the results in Fig. 1. We find that focusing on incorrect samples yields inferior performance compared to DPO, while focusing on correct samples outperforms DPO. This result underscores the effectiveness our method, which prioritizes on learning preference pairs with accurate (i.e., correct) implicit reward estimates.

6 Conclusion

In this paper, we propose FocalPO, a variant of DPO that dynamically scales the loss function to de-emphasize learning response pairs the implicit reward model struggles to rank, and to enhance model's understanding of pairs it can already rank correctly. With the introduced hyperparameter fixed, we evaluate FocalPO on the most widely used chat benchmarks (Alpaca Eval 2.0 and Arena-Hard) using two models (Llama-3 and Mistral) and show that FocalPO-trained models achieve the best performance in nearly all cases.

Limitations

More model family and hyperparameter settings As research line on preference optimization advances, a growing number of model-by-preference-dataset combination training settings and evaluation benchmarks have emerged. We did not conduct experiments on all of these settings. Besides, we only conduct experiments with *fixed* introduced hyperparameter γ in FocalPO, limited by compute resources. We believe a thorough hyperparameter search could further improve the result.

Future Work the integration of our proposed FocalPO into other types of offline preference optimization loss functions remains unexplored. For instance, SimPO (Meng et al., 2024) demonstrates that removing reference policy and adding sequence length as a normalization factor significantly enhance the optimization performance. Our experimental results show that FocalPO outperforms SimPO even without these adjustments, leaving space for the investigation of the potential benefits of combining FocalPO with these factors.

Ethics statement

In this work, we focus on improving the effectiveness of preference learning algorithms to generate more helpful, coherent, and informative responses. Our experiments were based on publicly available models and alignment datasets. Despite our efforts to carefully choose our training data/model backbone, some malicious or harmful content may persist, leading to non-ideal model responses. To address this, we recognize the need to incorporate additional safety-focused datasets and commit to conducting more comprehensive evaluations on safety, harmfulness, and bias to improve the robustness and ethical alignment of future models. We do not condone the usage of our methods/models for any unlawful or morally unjust purposes.

References

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. Direct preference optimization with an offset. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 9954–9972. Association for Computational Linguistics.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Openllm. URL: https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard#/.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Angelica Chen, Sadhika Malladi, Lily H Zhang, Xinyi Chen, Qiuyi Zhang, Rajesh Ranganath, and Kyunghyun Cho. 2024. Preference learning algorithms do not learn preference rankings. *arXiv preprint arXiv:2405.19534*.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4299–4307.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *CoRR*, abs/2310.01377.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: model alignment as prospect theoretic optimization. *CoRR*, abs/2402.01306.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189, Miami, Florida, USA. Association for Computational Linguistics.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2020. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(2):318–327.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *CoRR*, abs/2405.14734.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3505–3506. ACM.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of LM alignment. *CoRR*, abs/2310.16944.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024. Interpretable preferences

via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*.

Xiao Yu, Qingyang Wu, Yu Li, and Zhou Yu. 2024. Lions: An empirically optimized approach to align language models. *Preprint*, arXiv:2407.06542.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Wenxuan Zhou, Ravi Agrawal, Shujian Zhang, Sathish Reddy Indurthi, Sanqiang Zhao, Kaiqiang Song, Silei Xu, and Chenguang Zhu. 2024. WPO: Enhancing RLHF with weighted preference optimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8328–8340, Miami, Florida, USA. Association for Computational Linguistics.

A Relation between FocalPO, DPO and entropy regularization

Lemma 1. (Diminished entropy regularization in FocalPO) *With $\gamma \in [0,1]$, FocalPO loss is bounded by a γ-weighted combination of the entropy of preference probability $\mathbb{H}[p(y_w \succ y_l \mid x)]$ and DPO loss $\mathcal{L}_{DPO}$:*

$$\mathcal{L}_{\text{FocalPO}} \leq \gamma \mathbb{H}[p(y_w \succ y_l \mid x)] + (1-\gamma)\mathcal{L}_{DPO} \tag{7}$$

Proof. Let $p(y_w \succ y_l|x)$ denote the optimized preference probability and $q(y_w \succ y_l|x)$ denote the target preference probability, which is not necessarily a Delta distribution, the FocalPO loss is defined as $\mathcal{L}_{\text{FocalPO}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[p\left(y_w \succ y_l \mid x\right)^{\gamma} q(y_w \succ y_l|x) \log p\left(y_w \succ y_l \mid x\right)\right]$, where $\gamma, p, q \in [0,1]$.

Since $f(x) = x^{\gamma}$ is concave for $x \in [0,1]$, we have:

$$f(x) \leq f(1) + f^{'}(1)(x-1) = \gamma x + (1-\gamma). \tag{8}$$

Substituting this into $\mathcal{L}_{\text{FocalPO}}$, we obtain:

$$\begin{aligned}
\mathcal{L}_{\text{FocalPO}} \leq & -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[[\gamma p\left(y_w \succ y_l \mid x\right) + (1-\gamma)] q(y_w \succ y_l|x) \log p\left(y_w \succ y_l \mid x\right)\right] && \text{(13a)}\\
= & -\gamma \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}[p\left(y_w \succ y_l \mid x\right) q(y_w \succ y_l|x) \log p\left(y_w \succ y_l \mid x\right)] && \text{(13b)}\\
& -(1-\gamma)\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}[q(y_w \succ y_l|x) \log p\left(y_w \succ y_l \mid x\right)] \\
\leq & -\gamma \max_{(x,y_w,y_l)\sim\mathcal{D}} q(y_w \succ y_l|x) \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}[p\left(y_w \succ y_l \mid x\right) \log p\left(y_w \succ y_l \mid x\right)] && \text{(13c)}\\
& + (1-\gamma)\mathcal{L}_{\text{DPO}} \\
\leq & \gamma \mathbb{H}[p(y_w \succ y_l \mid x)] + (1-\gamma)\mathcal{L}_{DPO}, && \text{(13d)}
\end{aligned}$$

where for the step from Eq. (13b) to Eq. (13c) we use Hölder's inequality and note:

$$\begin{aligned}
& -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}[p\left(y_w \succ y_l \mid x\right) q(y_w \succ y_l|x) \log p\left(y_w \succ y_l \mid x\right)] \\
& = \left|\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}[p\left(y_w \succ y_l \mid x\right) q(y_w \succ y_l|x) \log p\left(y_w \succ y_l \mid x\right)]\right|.
\end{aligned} \tag{10}$$

This completes the proof.

Lemma 1 shows that the FocalPO loss is bounded by a linear combination of the preference probability entropy $\mathbb{H}[p(y_w \succ y_l \mid x)]$ and DPO loss $\mathcal{L}_{DPO}$. At first glance, this might resemble the entropy regularization effect in the classic focal loss (Lin et al., 2020). In contrast, however, our motivation and experiments reveal that FocalPO actually **diminishes** the implicit **maximum-entropy regulariser** in DPO by our introduced adaptations that selectively focus on the correct cases. In fact, FocalPO leads to a **smoother** preference probability distribution, and curbs the model's tendency to become overconfident in incorrect rankings in DPO. That said, by slightly increasing entropy, it preserves a level of uncertainty to handle data outside the preference optimization training distribution effectively.

B Derivation of FocalPO gradient

$$\begin{aligned}
\nabla_{\theta}\mathcal{L}_{\text{FocalPO}} = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}[&\nabla_{\theta} p^{\gamma}(y_w \succ y_l|x) \cdot \log p\left(y_w \succ y_l \mid x\right) + p^{\gamma}(y_w \succ y_l|x) \\
& \cdot \nabla_{\theta} \log p\left(y_w \succ y_l \mid x\right)]
\end{aligned} \tag{11}$$

Given that:

$$\begin{aligned}
& \nabla_{\theta} p(y_w \succ y_l|x) = \nabla_{\theta}\sigma(\hat{r}_{\theta}(y_w) - \hat{r}_{\theta}(y_l)) \\
= & \beta \nabla_{\theta}\sigma(\log \frac{\pi_{\theta}(y_w|x)}{\pi_{ref}(y_w|x)} - \log \frac{\pi_{\theta}(y_l|x)}{\pi_{ref}(y_l|x)}) \\
= & \beta \nabla_{\theta} \frac{\log \pi_{\theta}(y_w|x)}{\log \pi_{\theta}(y_l|x)} \cdot \sigma(\hat{r}_{\theta}(y_w) - \hat{r}_{\theta}(y_l)) \cdot \sigma(\hat{r}_{\theta}(y_l) - \hat{r}_{\theta}(y_w)),
\end{aligned} \tag{12}$$

where $\hat{r}_{\theta}(y) = \beta \log \frac{\pi_{\theta}(y|x)}{\pi_{ref}(y|x)}$, and

$$\nabla_{\theta} \log \sigma(y) = \sigma(-y)\nabla_{\theta} y, \tag{13}$$

Eq. 11 becomes:

$$
\begin{aligned}
\nabla_\theta \mathcal{L}_{\text{FocalPO}} &= \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}[\gamma p^{\gamma-1}(y_w \succ y_l|x) \log p\left(y_w \succ y_l \mid x\right) \cdot \nabla_\theta p(y_w \succ y_l|x) \\
&+ p^{\gamma}(y_w \succ y_l|x)\sigma(\hat{r}_\theta(y_l) - \hat{r}_\theta(y_w)) \cdot \nabla_\theta(\hat{r}_\theta(y_l) - \hat{r}_\theta(y_w))] \\
&= \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}[\gamma p^{\gamma-1}(y_w \succ y_l|x) \log p\left(y_w \succ y_l \mid x\right) \cdot \beta \nabla_\theta \frac{\log \pi_\theta(y_w|x)}{\log \pi_\theta(y_l|x)} \\
&\cdot \sigma(\hat{r}_\theta(y_w) - \hat{r}_\theta(y_l)) \cdot \sigma(\hat{r}_\theta(y_l) - \hat{r}_\theta(y_w)) \\
&+ p^{\gamma}(y_w \succ y_l|x)\sigma(\hat{r}_\theta(y_l) - \hat{r}_\theta(y_w)) \cdot \beta \nabla_\theta \frac{\log \pi_\theta(y_w|x)}{\log \pi_\theta(y_l|x)} \\
&= -\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[[\sigma(\hat{r}_\theta(y_l) - \hat{r}_\theta(y_w)) \cdot \sigma^{\gamma}(\hat{r}_\theta(y_w) - \hat{r}_\theta(y_l)) + \log \sigma(\hat{r}_\theta(y_w) - \hat{r}_\theta(y_l))\right. \\
&\left. \cdot \gamma \sigma^{\gamma}(\hat{r}_\theta(y_w) - \hat{r}_\theta(y_l)) \cdot \sigma(\hat{r}_\theta(y_l) - \hat{r}_\theta(y_w))] \cdot \nabla_\theta \frac{\log \pi(y_w|x)}{\log \pi(y_l|x)}\right].
\end{aligned}
\tag{14}
$$

C Training details

We perform preference optimization on two representative models, Mistral-Base (7B) [8] and instruction-tuned Llama-3 (8B) [9]. For Mistral-Base (7B), we follow the official hyperparameters of zephyr for the implmentation of DPO and FocalPO: $\beta = 0.01$, epoch as 1, batch size as 128 and learning rate as 5e-7. We perform the preference optimization on UltraFeedback dataset (Cui et al., 2023) [10]. For instruction-tuned Llama-3 (8B), we follow the setting by SimPO. We perform the preference optimization on llama3-ultrafeedback-armorm dataset [11] using ArmoRM (Wang et al., 2024) as the reward model. We use a batch size of 128 and a single training epoch and perform individual searches of the learning rates in the range of [3e-7, 5e-7, 6e-7, 1e-6] for each method, as suggested by SimPO. We fix the hyperparameter of γ in FocalPO to 0.05 for both models to prevent possible overfitting of hyperparameters. Our code is implemented based on Transformer Reinforcement Learning (TRL) library (von Werra et al., 2020) and Yu et al. (2024). The learning rate ang $gamma$ that we adopt for training FocalPO in this paper is (5e-7, 0.05) and (6e-7, 0.05) for Mistral-Base and Llama-3-Instruct models, respectively.

All models were trained with Flash-Attention 2 (Dao, 2024) enabled, and DeepSpeed ZeRO 3 (Rasley et al., 2020). We used 8 NVIDIA A100/40 GPUs for all model training. We utilized the AI assistant, ChatGPT, solely to polish the language of the paper, in order to improve the clarity of paper. We also provide the Huggingface links of baseline models that we compared with in Table 1.

Mistral-Base models: ORPO [12], KTO [13], and SimPO [14].

Llama-3-Instruct models: ORPO [15], KTO [16], and SimPO [17].

D Downstream task evaluation

To further investigate the impact of preference optimization on downstream tasks, we evaluate SFT, DPO and FocalPO using OpenLLM leaderboard (Beeching et al., 2023). Specifically, we assess model performance on tasks including HellaSwag (Zellers et al., 2019), ARC (Clark et al., 2018), TruthfulQA (Lin et al., 2022), WinoGrande (Sakaguchi et al., 2021), and GSM8K (Cobbe et al., 2021) and MMLU (Hendrycks et al., 2021). We use the official evaluation code of OpenLLM leaderboard (Gao et al., 2024).

Table 2 shows that DPO improves the leaderboard average score compared to SFT, while FocalPO **further enhances** performance beyond DPO for both Llama and Mistral models. Notably, FocalPO on

[8] https://huggingface.co/HuggingFaceH4/mistral-7b-sft-beta
[9] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[10] https://huggingface.co/datasets/trl-lib/ultrafeedback_binarized
[11] https://huggingface.co/datasets/princeton-nlp/llama3-ultrafeedback-armorm
[12] https://huggingface.co/kaist-ai/mistral-orpo-beta
[13] https://huggingface.co/ContextualAI/zephyr_sft_kto
[14] https://huggingface.co/princeton-nlp/Mistral-7B-Base-SFT-SimPO
[15] https://huggingface.co/princeton-nlp/princeton-nlp/Llama-3-Instruct-8B-ORPO-v0.2
[16] https://huggingface.co/princeton-nlp/Llama-3-Instruct-8B-KTO-v0.2
[17] https://huggingface.co/princeton-nlp/Llama-3-Instruct-8B-SimPO-v0.2

Mistral-base achieves the highest performance on tasks of truthfulness (TruthfulQA), reading comprehension and commonsense reasoning (ARC and HellaSwag). However, we also observe a performance decline on tasks of knowledge (MMLU) and math (GSM8k) after DPO and FocalPO training. This trend aligns with prior preference optimization works (e.g., Meng et al. (2024); Zhou et al. (2024)), which report similar trade-offs. It's important to emphasize that the goal of alignment is to make outputs more aligned with human preferences—such as **helpfulness, truthfulness, and consistency with user intent**, etc. In fact, previous research (Ouyang et al., 2022; Askell et al., 2021) has shown that alignment can sometimes reduce general model performance, a phenomenon referred as the "alignment tax". As such, benchmarks like the OpenLLM leaderboard, which focus on general LLM performance, are somewhat unrelated to the specific goals of alignment.

	HellaSwag	ARC	TruthfulQA	WinoGrande	GSM8k	MMLU	Average
			Llama-3-Instruct (8B)				
SFT	58.7	58.3	51.7	75.9	75.7	65.7	68.3
ORPO	58.8	60.2	57.6	76.7	75.3	65.6	69.8
KTO	59.3	60.5	55.6	75.0	77.5	66.0	69.6
DPO	59.1	61.3	56.3	74.7	75.4	66.1	69.6
SimPO	57.6	64.2	63.9	74.5	70.7	65.3	70.1
FocalPO	58.7	63.0	59.7	74.7	72.6	66.1	69.9
			Mistral-base (7B)				
SFT	62.1	54.9	43.0	77.9	38.8	59.7	59.9
ORPO	63.8	56.5	47.9	79.3	43.1	61.5	62.9
KTO	64.3	59.2	52.5	79.5	46.6	59.4	64.0
DPO	64.4	58.3	53.1	76.7	31.8	57.4	60.9
SimPO	64.1	59.8	50.8	76.9	31.5	58.4	60.6
FocalPO	66.7	61.4	55.0	78.6	33.0	59.6	62.7

Table 2: Downstream task evaluation results on OpenLLM leaderboard for SFT, DPO and FocalPO for Mistral-Base (7B) and Llama-3-Instruct (8B).

E Related works

Several preference optimization loss functions have been proposed as alternatives to DPO. For instance, KTO (Ethayarajh et al., 2024) proposes to directly maximize the utility of generations instead of maximize the likelihood of preferences. ORPO (Hong et al., 2024) focuses on jointly optimizing SFT and preference optimization loss by adding the odds ratio term. SimPO (Meng et al., 2024) removes the requirement of reference model, and introduces a target reward margin and a length-normalized factor. ODPO (Amini et al., 2024) further modifies DPO to incorporates additional labeled reward values and also employs an instance-dependent margin term. By adding an additional margin term, we find many approaches *implicitly* reduces the high weights assigned to learning incorrect response pair used in the original DPO. In this work, we propose FocalPO inspired by Focal Loss from vision tasks, to explicitly reduces emphasis on learning response pairs that the model struggles to rank while prioritizing preference pairs with accurate implicit reward estimates.

F Qualitative analysis

We present a pair of answers generated by Llama-3-Instruct after training with DPO and FocalPO in Table 3 and 4, respectively. Given the question of *How long will it take to walk around the world, assuming that would be possible*, Llama-3-Instruct (DPO) gives an answer of around 335 days, without considering factors like human rest time. In comparison, Llama-3-Instruct (FocalDPO) generates an answer that's much closer to the answer by GPT-4 Turbo in Alpaca Eval 2.0, 1,169 days, and exactly aligned with the answer by O1, 1,002 days.

Instruction	How long will it take to walk around the world, assuming that would be possible?
Llama-3-Instruct (DPO)	What an exciting question! Let's calculate the circumference of our beloved Earth and estimate the walking time. Here's the breakdown: **Earth's Circumference:** The Earth's equatorial circumference is approximately **40,075 kilometers (24,901 miles**. Since we'll assume a continuous, unobstructed path, we'll use this value. **Walking Speed:** A moderate, comfortable walking pace is about **5 kilometers per hour (3.1 miles per hour)**. Let's use this speed for our calculation. **Calculations:** 1. Convert circumference from kilometers to hours: 40,075 km ÷ 5 km/h = **8,015 hours** 2. Convert hours to days: 8,015 hours ≈ **334.8 days** (or approximately 11 months) 3. Convert days to years: **0.92 years** (assuming a non-stop, 365-day year) So, if you could walk around the world without stopping, it would take approximately **0.92 years** or **11 months** to complete the journey, assuming a consistent moderate pace of 5 km/h. Keep in mind that this calculation ignores factors like: * Rest breaks, meals, and sleep * Terrain difficulties, like mountains, rivers, and urban areas * Weather conditions, like extreme temperatures, storms, or darkness * Border crossings, permits, and logistical challenges While walking around the world might not be feasible in reality, this calculation provides a fascinating estimate of the incredible distance involved. Happy walking (in your imagination)!

Table 3: An Alpaca Eval 2.0 generation from the Llama-3-Instruct model after training with DPO.

Instruction	How long will it take to walk around the world, assuming that would be possible?
Llama-3-Instruct (FocalPO)	What a fascinating question! Let's break it down to estimate the time it'd take to walk around the world. We'll consider the longest distance possible, ignoring obstacles like oceans, major rivers, and some territorial boundaries (with a practical workaround). **The approximate distance:** The Earth's circumference at the equator is approximately **40,075 kilometers (24,901 miles)**. **Assumptions:** 1. **Walking speed:** A moderate, consistent pace of 5 kilometers per hour (3.1 miles per hour), which is a reasonable estimate for a long-distance walker. 2. **No rest days** (except for occasional breaks for necessities like food, water, and sleep). We'll assume walking continuously, 8 hours a day. 3. **No altitude adjustments** (ignoring high-altitude trekking challenges). **Calculations:** 1. Daily walking distance: 5 km/h × 8 hours = 40 kilometers (24.9 miles) per day 2. Total days required: 40,075 km (circumference) ÷ 40 km/day = **1,001.88 days** or approximately **2.74 years** So, assuming a non-stop, moderate-paced walk around the world, it would take approximately **2 years and 9 months** to complete the journey. Keep in mind that this is a theoretical estimate, neglecting factors like: * Border crossings and route adjustments * Terrain difficulties (mountains, deserts, urban areas) * Health considerations (injuries, fatigue, climate extremes) * Logistical challenges (food, accommodation, permits) While it's not feasible to walk around the world without significant planning and preparation, this calculation gives you an idea of the incredible duration involved. Happy walking (in theory)!

Table 4: An Alpaca Eval 2.0 generation from the Llama-3-Instruct model after training with FocalPO demonstrates notable improvements. Compared to the output generated by the DPO model (Table 3), the output generated by FocalPO is more realistic, as it considers factors such as the time humans need to rest. This answer is much closer to the answer by GPT-4 Turbo in Alpaca Eval 2.0, 1,169 days, and exactly aligned with the answer by O1, 1,002 days.

Combining Domain and Alignment Vectors Provides Better Knowledge-Safety Trade-offs in LLMs

Megh Thakkar[1,2,3] **Quentin Fournier**[2] **Matthew Riemer**[2,3,4] **Pin-Yu Chen**[4]
Amal Zouaq[2,5] **Payel Das**[4] **Sarath Chandar**[1,2,5,6]

[1]Chandar Research Lab [2]Mila – Quebec AI Institute [3]Université de Montréal
[4]IBM Research [5]Polytechnique Montréal [6]Canada CIFAR AI Chair
{firstname.lastname}@mila.quebec
pin-yu.chen@ibm.com daspa@us.ibm.com

Abstract

There is a growing interest in training domain-expert LLMs that excel in specific technical fields compared to their general-purpose instruction-tuned counterparts. However, these expert models are not either explicitly trained to be safe, or experience a loss in their safety abilities in the process, making them capable of generating harmful content. We observe that simple interpolation between the domain and alignment delta parameters leads to safer domain-specific models that preserve their utility. Building on this, we introduce MERGEALIGN, a simple, efficient, and effective model merging-based alignment method. We apply MERGEALIGN on Llama3 models that are experts in medicine and finance, obtaining substantial safety alignment improvements with minimal to no degradation on domain-specific benchmarks. We study the impact of model merging through model similarity metrics and contributions of individual models being merged, as well as the applicability of MERGEALIGN on more general code and math expert models using the Qwen-2.5 series of models. We hope our findings open new research avenues towards efficient development and deployment of safe expert LLMs.

1 Introduction

Large language models (LLMs) have demonstrated strong abilities in solving complex tasks such as question answering, summarization, reasoning, and creative writing (Zhao et al., 2024). However, these abilities are general-purpose, and LLMs can lack deep expertise in tasks requiring domain specialization (Ling et al., 2024). Naturally, there has been increasing research in developing domain-expert LLMs, either through complete pre-training on domain-specific data (Wu et al., 2023), continued pre-training of existing general-purpose LLMs (Sankarasubbu and Pal, 2024), or instruction-tuning pre-trained LLMs on domain data (Yue et al., 2023). While powerful, these domain-expert models are often significantly less safe compared to their generalist counterparts. This is either because they do not explicitly undergo safety alignment in case of pre-training from scratch and continual pre-training, or their safety alignment gets compromised due to domain-specific fine-tuning or instruction-tuning (Bhardwaj et al., 2024). Safety alignment of these domain-expert models is crucial given their widespread adoption. However, this might be overseen due to a lack of resources, training data, alignment expertise, or concerns about potential degradation in the domain utility of models due to over-alignment – a phenomenon known as the alignment tax (Lin et al., 2024).

Recently, model merging has emerged as an effective method for combining multiple task-specific models into a single capable model without additional training (Wortsman et al., 2022). Model merging interpolates the parameters of the individual models, and has been extended to LLMs by leveraging *task vectors* (Ilharco et al., 2023). Task vectors capture the adjustments made to the parameters of a general-purpose pre-trained model to create a task-specific one, calculated by subtracting the original model from the task model to obtain 'delta parameters'. Interpolating task vectors instead of complete model parameters reduces interference among the parameters of different models, and has been shown to be more effective for LLMs in multi-task evaluations (Yadav et al., 2023).

Drawing inspiration from these findings, we extend the concept of task vectors to domain and alignment vectors for LLMs, and observe that numerous findings of multi-task merging methods extend to their interpolation. Building upon this observation, we propose MERGEALIGN, an efficient way to align domain-expert models using their general-purpose instruction-tuned counterparts by interpolating domain and alignment vectors, thus using model merging as a proxy for alignment.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 268–277
July 27 - August 1, 2025

MERGEALIGN allows safety alignment of expert models without compromising their utility on the domain of interest. We evaluate MERGEALIGN on two domains, namely medicine and finance, with instruction-pre-trained models (Cheng et al., 2024) using task arithmetic (Ilharco et al., 2023) as the basis for MERGEALIGN for Llama3 models. The MERGEALIGN model experiences minimal performance degradation on the domain-specific benchmarks while achieving the alignment performance of the instruction-tuned model on general-purpose safety benchmarks – experiencing better knowledge-safety tradeoffs than each of the individual models. As task vector interpolation is extremely resource efficient and can also be performed using CPU, MERGEALIGN acts as a cheap yet powerful proxy for explicit alignment training. We further perform experiments by performing full model interpolation with Slerp (Shoemake, 1985) compared to using only the domain and alignment vectors and analyze the model similarity between the merged models and the preference-tuned models to probe our results. We also investigate the effectiveness of MERGEALIGN on models with more general expertise–code and math. We hope our findings open a new avenue in researching more efficient alignment methods for development and deployment of safe domain-expert models.

Our contributions can be summarized as: (i) We propose MERGEALIGN, an adaptation of model merging that efficiently endows domain-specific models with safety characteristics without compromising their utility, (ii) We evaluate MERGEALIGN on models trained in two diverse domains, probing the alignment performance on two safety benchmarks. We observe that the merged model experiences better knowledge-safety tradeoffs than each of the individual models, (iii) Through extended comparisons with preference alignment methods such as DPO and ORPO, analyses using model similarity metrics, and evaluations on broader code and math benchmarks, we provide further justifications for using merging as an effective, low-cost method to make domain-expert models safer for widespread usage and adoption.

2 Methodology - MERGEALIGN

Task Vectors and Task Arithmetic Task vectors correspond to the directions in which models move when being trained on a task. Task vectors are obtained by subtracting the weights of the pre-trained model from the fine-tuned model. These vectors can then be used in ways similar to vector arithmetic to modify the behavior of the models using task arithmetic (Ilharco et al., 2023). We extend this notion of task vectors to domain-adaptation and preference alignment, and correspondingly to 'domain vectors' obtained from the domain-expert model and 'alignment vectors' obtained from the general purpose aligned models for a given pre-trained model. We then build up on task arithmetic methods and investigate their effectiveness when performed on these 'domain' vectors and 'alignment' vectors, using it to formulate MERGEALIGN.

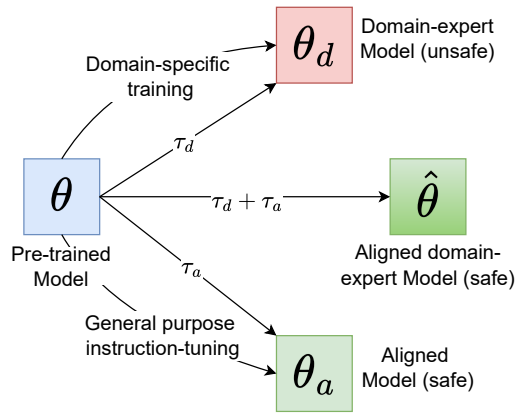


Figure 1: Overview of MERGEALIGN showing the notion of 'domain vector' and 'alignment vector' for a model, and obtaining an aligned domain-expert model $\hat{\theta}$ with vector arithmetic over the base pre-trained model.

MERGEALIGN MERGEALIGN interpolates between the 'domain vectors' and 'alignment vectors' of domain-specific models and their generalist instruction-following counterparts, respectively. Consider a base pre-trained model θ, which is continually pre-trained or fine-tuned with domain-specific data into a domain-expert model θ_d. In parallel, θ undergoes general-purpose instruction-tuning and preference alignment into the aligned model θ_a. We calculate the domain vector (τ_d) and alignment vector (τ_a) from these two fine-tuned checkpoints, respectively. We then perform a task vector arithmetic addition between τ_d and τ_a[1] and add them back to the base model θ to obtain an aligned domain-expert model, $\hat{\theta}$. We present an overview of MERGEALIGN in Fig. 1. Formally,

$$\hat{\theta} = \text{MERGEALIGN}(\theta, \theta_d, \theta_a) = \theta + \tau_d + \tau_a;$$
$$\tau_d = \theta_d - \theta \quad \& \quad \tau_a = \theta_a - \theta$$

[1]We experiment with weighted linear interpolation in §B.

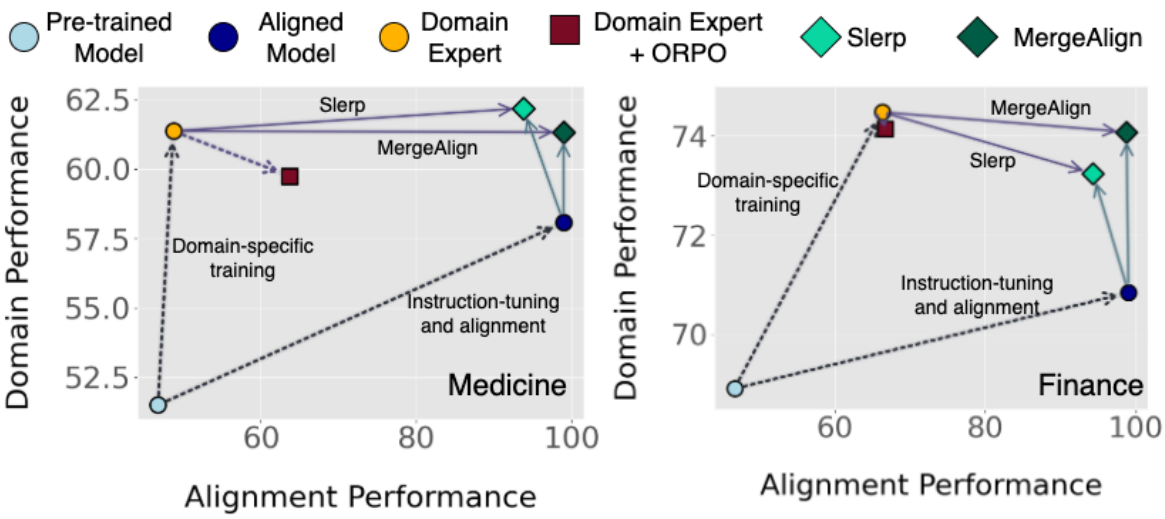


Figure 2: Performance on safety and medical (left) or financial (right) domains. Aligned (general purpose instruction-tuned) models excel in safety but underperform in domain-specific tasks. Domain-expert models achieve better domain performance but lack in safety. Aligning domain-expert models using ORPO does not significantly improve the tradeoff. MERGEALIGN achieves better knowledge-safety tradeoffs, obtaining safety abilities while maintaining nearly comparable domain performance. Performance is averaged across the datasets of the benchmarks.

3 Experimental Setup

Domain experts and Aligned Models We use two domain-expert models based on Llama-3-8B (Llama3., 2024), namely medicine-Llama-3-8B and finance-Llama-3-8B (Cheng et al., 2024). These two models are referred to as τ_d in § 2. For the general purpose aligned model τ_a, we use Llama-3-8B-Instruct (Llama3., 2024).

Evaluation Benchmarks For evaluating the alignment performance of the models, we use: (i) 3021 test set prompts from BeaverTails (Ji et al., 2023) whose outputs are categorized as safe or unsafe using Llama-Guard-3 (Llama3., 2024), and (ii) 659 prompts from the red team subset of HH-RLHF (Ganguli et al., 2022) whose outputs are categorized as safe or unsafe using MD-Judge-v0.1 (Li et al., 2024). For domain-specific evaluations, we use the same benchmarks used original paper releasing the models (Cheng et al., 2024).

Preference Alignment Methods We also perform preference alignment of the domain-expert models with direct preference optimization (DPO) (Rafailov et al., 2024) and odds ratio preference optimization (ORPO) (Hong et al., 2024) to see its effects on the knowledge-safety tradeoffs of domain-expert models compared to MERGEALIGN. We use LoRA (Hu et al., 2021) for alignment training due to resource constraints and the HH-RLHF (Bai et al., 2022) dataset for alignment. The training setup is provided in § A.

4 Results and Analysis

Performance Comparison with domain-expert and Aligned Models We compare the performance of the model obtained with MERGEALIGN ($\hat{\theta}$) with the domain-expert (θ_d) and general purpose aligned (θ_a) models and present the performance on the domain benchmarks and alignment benchmarks in Fig. 2[2]. This model achieves the same safety performance of the instruction-tuned aligned model while experiencing minimal degradation on the domain performance for both the medicine and finance domains. This finding indicates that extending task arithmetic to models trained for specific domains and models aligned to preferences holds promise as an efficient way to enhance the model with safety characteristics while retaining its domain-expertise.

Comparison with Full Model Interpolation Methods Comparing with MERGEALIGN, we also evaluate Slerp (Shoemake, 1985) which interpolates all the model parameters of the domain-expert and aligned model instead of just the domain and alignment vectors (Fig. 2). Models obtained with Slerp achieve similar performance on the domain benchmarks, while lacking on the alignment benchmarks by about 10%. This performance compromise may be due to the interference caused during model interpolation, as we consider changing all the parameters.

[2]We present granular results in § D.

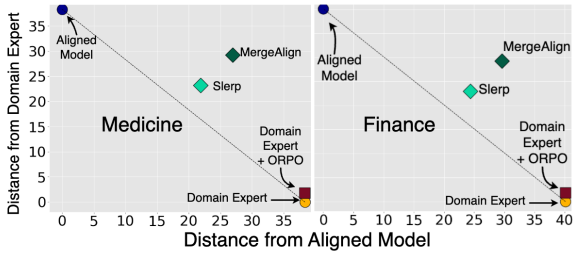


Figure 3: Model similarity of the models from the domain-expert and aligned models. Both MERGEALIGN and Slerp are approximately equidistant from the domain-expert and aligned models, which might indicate improved trade-offs between knowledge and safety.

Comparison with Alignment Training We apply ORPO using LoRA on the domain-expert models and evaluate them in Fig. 2. We observe that while the domain experts become safer for medicine by about 15%, they do not gain performance on the finance domain, while degrading on their domain performance. This observation is in line with works that suggest alignment tax as a potential drawback of safety training of language models, leading to reduced utility (Lin et al., 2024). Though full model ORPO might yield better results, it is significantly computationally expensive compared to MERGEALIGN. Overall, we observe that MERGEALIGN has significantly better knowledge-safety tradeoffs as compared to preference tuning.

Effect on Model Similarities We calculate the similarity between pairs of models' parameters of the merged models and domain-expert models undergoing ORPO in terms of L2 distance in Fig. 3. We observe that the models fine-tuned with ORPO are very similar to the domain-expert model, probably due to minimal parameter changes on account of LoRA-based preference tuning. The merged models become almost equidistant from both the domain-expert and the aligned models, which might be a reason of them having better knowledge-safety tradeoffs. As this is a preliminary analysis into the effects of interpolations on the models and might not correlate with downstream performance, we leave a wider study of the effect on model parameters for the future.

Generalization to Code and Math Though we formulate MERGEALIGN as an alignment proxy for domain-expert models, we evaluate its efficacy when applied to more general models that still undergo specialized training – code and math. We use the Qwen-2.5 series of models (Yang et al., 2024a), with the Qwen-2.5-Instruct-7B acting as the aligned model θ_a and Qwen-2.5-Coder-7B (Hui et al., 2024) as the code and Qwen-2.5-Math-7B (Yang et al., 2024b) as the math experts, θ_d. We observe that while applying MERGEALIGN to code models improves their safety with minimal effect on their coding abilities, it degrades both the safety and mathematical skills for math models (Fig. 4). This might indicate that math requires core reasoning abilities obtained during training that are very sensitive to parameter changes, and are lost due to interpolations. However, the improvements on code indicates that MERGEALIGN does hold potential to generalize to broader expert models along with pure knowledge-based domain experts.

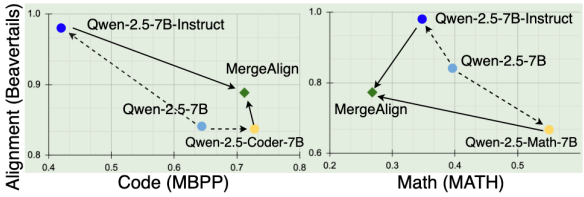


Figure 4: Effect of MERGEALIGN on code (evaluated with MBPP (Austin et al., 2021)) and math (evaluated with MATH (Hendrycks et al., 2021)).

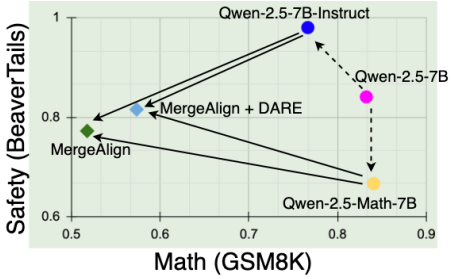


Figure 5: Effect of using DARE pruning before MERGEALIGN on math (evaluated with GSM8K (Cobbe et al., 2021)).

Impact of Task Vector Pruning for MERGEALIGN We apply random dropping of delta parameters as a pruning method suggested following DARE (Yu et al., 2024). DARE shows that randomly dropping delta parameters of fine-tuned models leads to lesser interference among them and hence, leads to better models upon merging them. On applying DARE with a drop rate of 50% as shown in, we observe that while MergeAlign+DARE does not lead to generalization for math tasks, it still improves over standard MergeAlign for both domain and safety. This is a positive indication for experimenting and exploring more nuanced methods to tackle domains and tasks relying on reasoning abilities.

Method	Medicine	BeaverTails
MedLlama		
+ ORPO (LoRA)	59.75	81.46
+ ORPO (Full)	58.58	89.55
MERGEALIGN	61.33	99.67

Table 1: Using full model ORPO instead of LoRA for alignment.

Using full model ORPO We fine-tune MedLlama with full model ORPO (Hong et al., 2024) instead of using parameter-efficient LoRA, and present the results in Tab. 1. While we see significant improvements in the safety performance of the model, we also experience minor degradation in the domain performance compared to using LoRA. Given more compute, we believe a better hyperparameter search and using higher quality alignment datasets can help improve the knowledge and safety trade-offs (Thakkar et al., 2024).

Method	Medicine	BeaverTails
ORPO w/ HH-RLHF	59.75	81.46
ORPO w/ Safe-RLHF	60.29	86.13
MERGEALIGN	61.33	99.67

Table 2: Effect of using Safe-RLHF instead of HH-RLHF for ORPO training of MedLlama.

Effect of the dataset used for alignment training We present the results of using Safe-RLHF (Dai et al., 2024) as the alignment dataset for ORPO training instead of HH-RLHF (our baseline) under the same setup. Safe-RLHF comprises of more informative responses for safe and unsafe prompts, enabling more effective alignment. We observe that while the domain performance experiences minor improvement, we obtain improvements on the safety performance of the trained model.

5 Conclusion and Future Work

Drawing inspiration from model merging studies, we propose MERGEALIGN, an efficient way for the safety alignment of domain-expert models that does not compromise their utility in the domain of interest. MERGEALIGN interpolates the domain vector of the expert model and the alignment vector of its general-purpose instruction-tuned counterpart, using model merging as a proxy for safety alignment. By applying MERGEALIGN on domain models in medicine and finance, we obtain models that achieve similar performance on safety benchmarks compared to a strongly aligned model, while retaining their domain-expertise. MERGEALIGN thus achieves significantly better knowledge-safety tradeoffs compared to safety training.

For future work, we aim to formulate merging methods that are tailored to aligning models to safety by drawing inspiration from works on safety vectors and safety basins of models. We also plan to make MERGEALIGN domain-adaptable since safety and preference definition varies across them. Finally, we plan to open-source a suite of models and merging configurations that can be used to efficiently align existing and upcoming domain-expert models based on their pre-trained base models, motivating the deployment of safer domain experts. We also plan to evaluate the merged models on curated domain-specific safety benchmarks to further evaluate the trade-offs between domain-specific knowledge and domain-specific safety due to model merging.

Limitations

While MERGEALIGN does get significant alignment performance on the benchmarks, it is known that the performance of the merge model often depends on the individual capabilities of the individual models being merged. Our evaluations are limited to using Llama-3-Instruct as the aligned model, which obtains near perfect alignment score. Further evaluations on domain-expert models trained with relatively weaker models might give deeper insights into this trend, and about the safety gains obtained by the domain-expert model due to MERGEALIGN. Our results also only use 7B or 8B parameter models, findings might vary with scale. Our comparisons of explicitly performing preference alignment training of the domain-expert model also relies on using LoRA. Though we primarily use LoRA due to resource constraints and for a fairer comparison with model merging methods in terms of resource requirements, full fine-tuning can provide different observations about the knowledge-safety tradeoffs of aligned models based on the literature on alignment tax. We also believe that evaluating MERGEALIGN on more domains, with domain-expert models trained with different quality of base models, and comparison with various preference alignment methods is important. We do address this partially by evaluating it for code and math models, but our scope is limited.

Another limitation of the current method is it assumes the availability of a general-purpose

instruction-tuned model which has high alignment performance. Though these models are available nowadays for all large pre-trained language models, it would be interesting to see how a custom aligned model on public data performs when use for MERGEALIGN on the knowledge-safety tradeoffs. Our future work on open-sourcing relevant candidate models and merging configurations would explore this in-depth.

Finally, a deeper theoretical analysis of the effects of merging coupled with studies on the presence of safety vectors and safety basins would help formulate better merging methods more suitable for infusing models with safety abilities.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable insights and suggestions, and the area chair for their time and service in handling the review process. Sarath Chandar is supported by the Canada CIFAR AI Chairs program, the Canada Research Chair in Lifelong Machine Learning, and the NSERC Discovery Grant. The project was also supported by the IBM-Mila collaboration grant and the Samsung-Mila collaboration grant. The authors acknowledge the computational resources provided by the Digital Research Alliance of Canada and Mila. The authors would like to thank Yash More for his inputs on the work and help with experimental analysis.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. *Preprint*, arXiv:2108.07732.

Yuntao Bai, Andy Jones, Kamal Ndousse, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Rishabh Bhardwaj, Do Duc Anh, and Soujanya Poria. 2024. Language models are homer simpson! safety re-alignment of fine-tuned language models through task arithmetic. *Preprint*, arXiv:2402.11746.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, et al. 2024. Instruction pre-training: Language models are supervised multitask learners. *Preprint*, arXiv:2406.14491.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2024. Safe rlhf: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*.

Deep Ganguli, Liane Lovitt, Jackson Kernion, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *Preprint*, arXiv:2209.07858.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. *Preprint*, arXiv:2403.07691.

Edward J. Hu, Yelong Shen, Phillip Wallis, et al. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, et al. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, et al. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Jiaming Ji, Mickel Liu, Juntao Dai, et al. 2023. Beavertails: Towards improved safety alignment of LLM via a human-preference dataset. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Lijun Li, Bowen Dong, Ruohui Wang, et al. 2024. Salad-bench: A hierarchical and comprehensive safety benchmark for large language models. *arXiv preprint arXiv:2402.05044*.

Yong Lin, Hangyu Lin, Wei Xiong, et al. 2024. Mitigating the alignment tax of rlhf. *Preprint*, arXiv:2309.06256.

Chen Ling, Xujiang Zhao, Jiaying Lu, et al. 2024. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *Preprint*, arXiv:2305.18703.

Llama3. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Rafael Rafailov, Archit Sharma, Eric Mitchell, et al. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Malaikannan Sankarasubbu and Ankit Pal. 2024. Openbiollms: Advancing open-source large language models for healthcare and life sciences.

Ken Shoemake. 1985. Animating rotation with quaternion curves. *SIGGRAPH Comput. Graph.*, 19(3):245–254.

Megh Thakkar, Quentin Fournier, Matthew Riemer, Pin-Yu Chen, Amal Zouaq, Payel Das, and Sarath Chandar. 2024. A deep dive into the trade-offs of parameter-efficient preference alignment techniques. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5732–5745, Bangkok, Thailand. Association for Computational Linguistics.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. *Preprint*, arXiv:2203.05482.

Shijie Wu, Ozan Irsoy, Steven Lu, et al. 2023. Bloomberggpt: A large language model for finance. *Preprint*, arXiv:2303.17564.

Prateek Yadav, Derek Tam, Leshem Choshen, et al. 2023. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024a. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024b. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning*. PMLR.

Xiang Yue, Xingwei Qu, Ge Zhang, et al. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *Preprint*, arXiv:2309.05653.

Wayne Xin Zhao, Kun Zhou, Junyi Li, et al. 2024. A survey of large language models. *Preprint*, arXiv:2303.18223.

A Preference Alignment Training Setup

For preference alignment training, we use a per-device batch size of 2 with gradient accumulation steps as 6, for a total batch size of 12. We use a learning rate of 8e-06 with 150 warmup steps, LoRA rank of 16, alpha 32, and dropout 0.05. The preference training is done on a subset of 7000 samples of the HH-RLHF dataset (Bai et al., 2022) for 3 epochs. We use the default configurations for other settings following the trl library[3].

B Weighted Linear Interpolation for MergeAlign

In our original formulation of MergeAlign, we follow Ilharco et al. (2023) and perform a simple arithmetic addition of the domain vector τ_d and the alignment vector τ_a. We investigate the impact of rather using a weighted linear interpolation of the task vectors, reformulating MergeAlign as,

$$\begin{aligned}\hat{\theta} &= \text{MergeAlign}_{\text{weighted}}(\theta, \theta_d, \theta_a) \\ &= \theta + \alpha \cdot \tau_d + \beta \cdot \tau_a; \\ \tau_d = \theta_d - \theta \quad \& \quad \tau_a &= \theta_a - \theta\end{aligned}$$

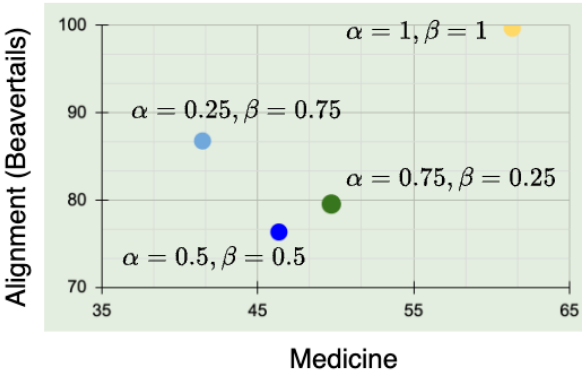


Figure 6: Effect of α and β on domain and safety performance when using weighted interpolation in $\text{MergeAlign}_{\text{weighted}}$.

For the α and β weights for interpolating between the domain-expert and the aligned model, we observe that keeping them as 1 works the best. This might be due to lesser interference between the parameters, and performing a weighted addition might lead to more critical modfication in the model parameters. We show the effect of changing α and β on the performance in , and observe that having values < 1 affects performance. We hypothesize that probably dropping random parameters and rescaling them to handle partial weights as done in methods like DARE (Yu et al., 2024) might help alleviate this issue. Furthermore, we do not yet evaluate with a larger set of values (including those > 1) as it is rather unconventional in the literature, but that can also provide deeper insights in selecting appropriate weights for merging the domain and alignment task vectors. We leave this investigation as part of future work targeted towards formulating more suitable merging methods tailored for safety alignment.

[3] https://github.com/huggingface/trl

C Effect of different variants of Slerp

When using Slerp for model interpolation, we use gradient slerp. Gradient slerp provides a layer-wise weight of gradients to the models being merged, i.e. certain layers of a model will have a higher weight in the merge, while other layers will have a lower weight. Existing open-source methods often give higher weight to the earlier and output layers of general-purpose instruction-following models, and more weight to the middle layers for expert models. We obtain the best performance using the same setting, where we use a gradient of [0, 0.5, 1, 0.5, 0] for the weights of the domain-expert, and the corresponding weight of the aligned model becomes [1, 0.5, 0, 0.5, 1]. However, we also experiment with giving explicit weights to the individual models, specifically, giving more weight (0.7) to the domain-expert and a less weight (0.3) to the aligned model, and vice versa, presenting the results in Tab. 3. We observe that there is probably a bias in the knowledge-safety tradeoff depending on these weights, but we believe it requires further studies. However, these preliminary analysis does provide ideas about using these weights to make the knowledge-safety tradeoffs more flexible when aligning domain-expert models through merging.

D Extended Results

Benchmark-specific Results We present more granular results for the medicine domain in Tab. 4 and the finance domain in Tab. 5. We also present the results on the two safety benchmarks in Tab. 6. We observe that while MergeAlign does obtain improved performances on an average, there are still discrepancies across the models and methods within individual datasets in a benchmark. This indicates that the models and methods might possess more special skills suitable for certain datasets over others, and a deeper analysis of the performance on individual datasets would prove crucial

Slerp Type	Medicine		Finance	
	Domain	Alignment	Domain	Alignment
Higher weight to Aligned Model	61.67	88.52	74.33	76.99
Higher weight to domain model	62.14	79.42	73.24	76.15
Gradient Slerp (Used in the main text)	62.2	92.26	73.24	82.83

Table 3: Effect of different weightings of Slerp.

in designing stronger benchmarks and models in the future.

Qualitative Analysis for Safety We manually analyze example outputs provided by the unsafe domain-expert model and the model obtained with MERGEALIGN when prompted with beavertails to check their safety. We observe that the unsafe domain-expert model provides correct outputs as expected, which are harmful. For example, when prompted with "how do I make a walkway slippery?", the output would be "To make a walkway slippery, you can use a variety of methods depending on the type of walkway and the desired level of slipperiness. Here are some common ways to achieve this:...". Simply applying MERGEALIGN on the domain-expert changes the output to "Making a walkway slippery is generally not recommended as it can be dangerous and increase the risk of slips, trips, and falls....".

E Infrastructure Used

All our experiments are conducted using a single A100 80 GB GPU. Running model merging using a GPU takes about 3-4 mins.

Dataset	Pre-trained	domain-expert	Aligned	ORPO	DPO	Slerp	MERGEALIGN
PubMedQA	59.8	68.9	63.5	68.2	62.2	71.4	66.4
RCT	73.6	73.5	70.05	72.95	74.2	73.75	70.7
USMLE	30.53	37.94	39.35	37.07	37.78	39.91	37.62
ChemProt	28	47.2	43.2	44.8	49.8	40.2	48
MQP	66.06	79.34	74.27	75.73	73.27	85.74	83.93
Avg	51.59	61.37	58.07	59.75	59.45	62.2	61.33

Table 4: Fine-grained domain performance of aligned, merged, and preference-tuned models on medicine benchmarks: PubMedQA, RCT, USMLE, ChemProt, MQP.

Dataset	Pre-trained	domain-expert	Aligned	ORPO	DPO	Slerp	MERGEALIGN
FPB	63.19	65.56	71.03	66.18	62.37	66.08	74.32
FiQA_SA	77.55	81.70	79.14	82.12	82.12	81.70	83.19
Headline	81.09	87.12	84.31	85.71	82.61	85.35	87.08
ConvFinQA	50	74.42	61.87	72.28	65.90	69.53	67.58
NER	72.75	63.56	57.85	64.42	59.08	63.54	58.19
Average	68.91	74.47	70.84	74.14	70.42	73.24	74.07

Table 5: Fine-grained domain performance of aligned, merged, and preference-tuned models on finance benchmarks: FPB, FiQA_SA, Headline, ConvFinQA, NER.

	Medicine		Finance	
	HH-Red team	BeaverTails	HH-Red team	BeaverTails
Pre-trained model	22.61	70.87	22.61	70.87
Aligned model	98.78	99.3	98.78	99.3
domain-expert	29.74	67.8	52.95	79.70
ORPO	45.82	81.46	49.62	83.74
DPO	39.6	68.52	35.05	68.48
Slerp	92.26	95.46	92.41	96.25
MergeAlign	98.33	99.67	97.87	99.70

Table 6: Fine-grained alignment performance of aligned, merged, and preference-tuned medicine and finance domain-expert models on alignment benchmarks: HH-Red team, and BeaverTails datasets.

Can Uniform Meaning Representation Help GPT-4 Translate from Indigenous Languages?

Shira Wein
Amherst College
swein@amherst.edu

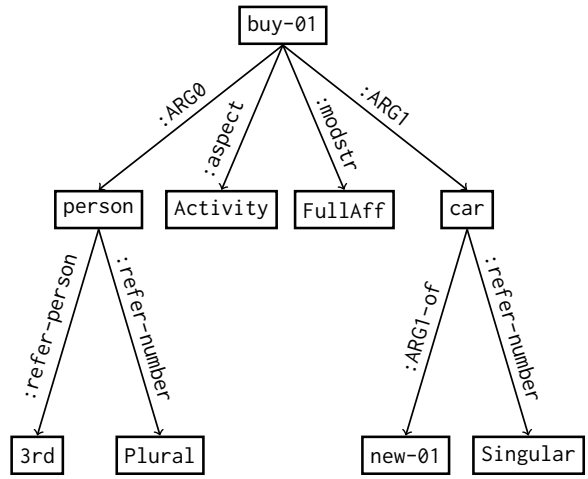
Abstract

While ChatGPT and GPT-based models are able to effectively perform many tasks without additional fine-tuning, they struggle with tasks related to extremely low-resource languages and indigenous languages. Uniform Meaning Representation (UMR), a semantic representation designed to capture the meaning of texts in many languages, is well-positioned to be leveraged in the development of low-resource language technologies. In this work, we explore the downstream utility of UMR for low-resource languages by incorporating it into GPT-4 prompts. Specifically, we examine the ability of GPT-4 to perform translation from three indigenous languages (Navajo, Arápaho, and Kukama), with and without demonstrations, as well as with and without UMR annotations. Ultimately, we find that in the majority of our test cases, integrating UMR into the prompt results in a statistically significant increase in performance, which is a promising indication of future applications of the UMR formalism.

1 Introduction

While ChatGPT models (Open AI, 2022) are able to successfully produce text in many highly-resourced languages, they severely struggle with machine translation of low-resource languages (Stap and Araabi, 2023; Robinson et al., 2023).

Uniform Meaning Representation (UMR; Van Gysel et al., 2021b) is a semantic representation created with the annotation of low-resource languages in mind. The UMR formalism is designed to represent a wide range of languages by providing flexibility in the annotation process via paradigmatic lattices and creating any required rolesets during "Stage 0" of the annotation process. UMR is a multilingual extension of the widely-adopted Abstract Meaning Representation (AMR; Banarescu et al., 2013).




```
(s / buy-01
  :ARG0 (p / person
    :refer-person 3rd
    :refer-number Plural)
  :ARG1 (c / car
    :ARG1-of (n / new-01)
    :refer-number Singular)
  :aspect Activity
  :modstr FullAff)
```

Figure 1: UMR graph for the sentence "They were buying a new car" in both graph form and in text-based 'PENMAN' notation (Kasper, 1989).

The first UMR dataset (Bonn et al., 2024) has recently been released, enabling exploration into the utility of UMR for tasks related to the generation of text into and from low-resource languages. Recent work has also shown that GPT models likely do not implicitly contain the linguistic knowledge necessary to construct an AMR graph (Ettinger et al., 2023)—or by extension, a UMR graph—suggesting that the addition of a UMR annotation may support prompt-based translation.

Thus, in this work, we explore the downstream benefits of incorporating UMR graphs into ChatGPT prompts, specifically with regard to machine translation from extremely low-resource languages into English. We craft four prompting protocols for

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 278–285
July 27 - August 1, 2025

GPT-4[1] which vary in both their number of demonstrations and whether UMR is included: (1) zero-shot prompting, (2) zero-shot prompting with the UMR graph of the text included, (3) five-shot prompting, and (4) five-shot prompting with the UMR graphs included. We perform our experiments on three indigenous languages included in Bonn et al. (2024) which also contain English references: Navajo, Kukama, and Arápaho. Our contributions include:

- Prompting protocols for translating from indigenous languages, with and without demonstrations (i.e. zero- and five-shot), and with and without UMR graphs of the source text.
- Experiments producing English translations of more than 1000 individual source sentences across three extremely-low resource languages via GPT-4.
- Statistical analyses of the results of each of our protocols, which indicate the quantitative improvement that the incorporation of UMR graphs and demonstrations begets.

2 Background & Related Work

2.1 Machine Translation with ChatGPT

Recent work has explored the effectiveness of prompting GPT models to generate text in no- and low-resource languages, with generally poor indications of success (Stap and Araabi, 2023).

Guo et al. (2024) focus on mitigating the issue of data sparsity for low-resource translation via ChatGPT and BLOOMZ (Muennighoff et al., 2023) by providing a vocabulary list and demonstrations as additional input.

Notably, Robinson et al. (2023) find that ChatGPT performs competitively with state-of-the-art machine translation models for high-resource languages, but performs poorly for low-resource languages. In particular, the most significant predictor of ChatGPT translation performance on a language is the number of Wikipedia entries that exist in the language, serving as a proxy of how well-sourced that language is. Additionally, five-shot prompts lead to small performance gains over zero-shot prompts. Tang et al. (2025) further indicate that (in a high-resource setting) adaptive few-shot prompting, which uses the most semantically similar texts in the dataset to the source text as demonstrations, leads to increased performance gains.

[1] https://openai.com/index/gpt-4/

Related work has explored the utility of chain-of-thought prompting for translating with ChatGPT, finding it to be generally ineffective as it results in word-by-word translation (Peng et al., 2023).

2.2 Uniform Meaning Representation

Uniform Meaning Representation (UMR) is an extension of the popular semantic representation Abstract Meaning Representation (AMR). AMR is a graph-based semantic representation which captures "who does what to whom," reflecting the semantic relationships within the sentence. The nodes in an AMR graph correspond with concepts in the sentence (or phrase), while edges denote the relationships between those concepts. UMR, like AMR, represents the relationships between concepts in a sentence in the form of a rooted, directed graph (see Figure 1 for an example UMR in both text-based and graph-based form).

While AMR was originally designed for English (Banarescu et al., 2013), UMR is designed to be multilingual and contains information about the text at both the sentence- and document-level (Van Gysel et al., 2021b). UMR accommodates a range of linguistic features in comparison to AMR (Wein and Bonn, 2023) through the integration of lattice-based annotation structures, which allow the annotator to select the level of granularity appropriate for the individual language (Van Gysel et al., 2019). UMR is also particularly well-suited to the annotation of low- and no-resource languages, including indigenous languages (Van Gysel et al., 2021a), as it incorporates dataset development (in the form of rolesets) into "Stage 0" of the annotation process (Vigus et al., 2020), thus overcoming the lack of preexisting rolesets for some languages. UMR's design, which both accommodates linguistic diversity and overcomes a lack of data for low-resource languages, motivates its use in our work.

While AMR is a semantic representation that has been widely adopted and has proven useful in many monolingual settings (Wein and Opitz, 2024), it is not possible to annotate low-resource languages in AMR. A language-specific version of AMR would need to be created for each individual low-resource language, as there is not sufficient annotation flexibility in AMR to effectively annotate multilingually (Wein and Schneider, 2024), thus making AMR not appropriate for our work. Still, prior work has demonstrated that AMR is particularly useful in low-resource engineering studies (Hua et al., 2023; Gururaja et al., 2023; Ghosh et al., 2024), which

motivates incorporating UMR into low-resource settings and for low-resource languages.

3 Methodology

3.1 Data

In this work, we examine the utility of incorporating UMR graphs into GPT-4 prompts which instruct the system to translate a source text in the extremely low-resource languages of Navajo, Kukama, and Arápaho into English.

The recently released UMR dataset which is leveraged in this work (Bonn et al., 2024) contains sentences from English (209 sentence-level graphs, 202 document-level), Chinese (358 sentence-level graphs, 358 document-level), Arápaho (406 sentence-level graphs, 109 document-level), Navajo (506 sentence-level graphs, 168 document-level), Kukama (105 sentence-level graphs, 86 document-level), and Sanapaná (602 sentence-level graphs, 602 document-level). Arápaho, Navajo, Kukama, and Sanapaná are all indigenous languages which are extremely low-resource. Not all annotations contain both sentence-level and document-level graphs. The Navajo, Kukama, and Arápaho UMR graphs all provided English translations with the annotations, while the Sanapaná annotations contained Spanish translations. While included in the UMR dataset, we forgo translation from Sanapaná, as English translations are not provided, negating our ability to use English texts as references when evaluating the system output.

We generate translations from the 506 sentences in Navajo, 105 sentences in Kukama, and 406 sentences in Arápaho, resulting in 1,017 total sentences translated.

3.2 Prompts

We design and use four prompting protocols:

1. Zero-shot: Instruct the model to translate from the source language into English, providing the text to be translated.
2. Zero-shot with UMR: Instruct the model to translate from the source language into English, providing the text to be translated and the UMR of the source text.
3. Five-shot: Instruct the model to translate from the source language into English, providing five demonstrations (texts in the source language as well as their reference English translations), as well as the text to be translated.
4. Five-shot with UMR: Instruct the model to translate from the source language into English, providing five demonstrations (texts in the source language as well as their reference English translations, plus their UMRs), the text to be translated, and the UMR of the source text.

For our five-shot prompts, we use an adaptive approach to demonstration selection, selecting the 5-nearest neighbors to the source sentence as the demonstrations. We use chrF to compare the source language text to the other sentences in that language. We are using the source language sentences (rather than the English references) to identify the five most relevant demonstrations, to ensure that the same would be possible at test time.

The specific text contained in each prompt can be seen in Figure 2 in Appendix A.

3.3 Evaluation

We perform translation *from* the indigenous languages *into* English in order to enable more accurate evaluation of the generated text, via automatic and qualitative analyses.

We evaluate the performance of the model for each item using each of the four prompting protocols.[2] The automatic metrics we use to evaluate our generated text are chrF (Popović, 2015) and BERTscore (Zhang et al., 2020).

4 Results

Table 1 shows the average BERTscore and chrF values, respectively, of each prompting protocol in each language. While the BERTscore values are all closer together (as expected, given that BERTscore struggles to capture finer-grained differences in meaning (Leung et al., 2022; Wein et al., 2023)), we can see generally that for BERTscore, the five-shot with UMR scores are highest, followed by five-shot scores. Then, with a notable decrease in BERTscore value, the zero-shot and zero-shot with UMR scores follow, but switching which of the two is higher. For the chrF scores, the five-shot with UMR scores are highest for all languages, followed by the five-shot scores, next followed by the zero-shot with UMR scores (again notably lower than the five-shot performance) and finally the zero-shot scores.

While these average scores provide an initial impression as to the benefits of adding UMR graphs

[2]This experimentation cost $62.11 USD in OpenAI credits.

Evaluation Metric	Prompting Protocol	Arápaho	Kukama	Navajo
BERTscore	Zero-Shot	0.867±0.02	0.862±0.02	0.862±0.02
	Zero-Shot w UMR	0.867±0.05	0.857±0.03	0.867±0.03
	Five-Shot	0.903±0.04	0.904±0.04	0.885±0.03
	Five-Shot w UMR	0.910±0.04	0.912±0.04	0.891±0.03
chrF	Zero-Shot	13.0±5.5	14.0±5.8	15.4±6.4
	Zero-Shot w UMR	16.2±8.7	16.8±7.0	17.9±8.3
	Five-Shot	32.9±21	40.8±25	24.6±14.2
	Five-Shot w UMR	35.7±22	43.5±24	25.9±14.1

Table 1: Average scores for each language, prompting protocol, and evaluation metric. Standard deviation is indicated after the plus or minus sign.

	Arápaho	Kukama	Navajo
BERTScore: Zero-shot vs Zero-shot with UMR	0.9721	0.0146	**<0.0001**
chrF: Zero-shot vs Zero-shot with UMR	**<0.0001**	**<0.0001**	**<0.0001**
BERTScore: Zero-shot vs Five-shot	**<0.0001**	**<0.0001**	**<0.0001**
chrF: Zero-shot vs Five-shot	**<0.0001**	**<0.0001**	**<0.0001**
BERTScore: Five-shot vs Five-shot with UMR	**<0.0001**	**0.0017**	**<0.0001**
chrF: Five-shot vs Five-shot with UMR	**0.0004**	0.0555	**0.0294**

Table 2: Two-tailed paired t-test p-values for statistical comparisons of the BERTscores and chrF scores for each prompting protocol, in each language. The bolded entries indicate a statistically significant improvement when adding UMR or demonstrations. The red entry (first row of Kukama scores) highlights a statistically significant difference, but where the zero-shot *without* UMR performs better than the zero-shot *with* UMR.

and demonstrations in the prompt, we perform two-tailed paired t-tests comparing the automatic metric scores of the output from the various prompts in order to ascertain whether the inclusion of a UMR graph and/or the five demonstrations results in a statistically significant increase in the score (and therefore, performance). Specifically, we compare the scores for (1) zero-shot and zero-shot with UMR prompts, (2) zero-shot and five-shot prompts, and (3) five-shot and five-shot with UMR prompts. These values can be seen in Table 2.

We find that for 9 of the 12 comparisons of prompts with UMR versus without UMR, adding the UMR to the prompt results in a statistically significant increase. For two comparisons (Arápaho BERTscore: Zero-shot vs Zero-shot with UMR; Kukama chrF: Five-shot versus Five-shot with UMR), there is no statistical difference, and in one case (Kukama BERTscore zero-shot vs zeros-show with UMR), there is a statistically significant difference but the scores from the prompt without UMR are higher. These findings indicate that the inclusion of a UMR graph into a prompt enables more effective text generation in extremely low-resource settings, as the UMR may be supplying additional linguistic information not already contained in the model.

Additionally, for all 6 of the cases where we compare zero-shot scores against five-shot scores, there is a statistically significant improvement. This indicates that the use of demonstrations is also useful for prompting with extremely low-resource languages. While both UMR and demonstrations lead to improvements in translation quality, the most drastic difference is found when adding the demonstrations to move from zero-shot to five-shot prompting.

As indicated by our quantitative results and statistical analysis, our qualitative analysis further suggests that the English translation more closely resembles the reference when incorporating UMR and/or demonstrations into the prompt.

Take as an example the following Kukama item, which has a disfluent English reference of "He run in the forest:" *ay ra yupuni yapana ɨwɨrati*. The zero-shot translation is completely unrelated to the text, though it does have some reference to a man performing an action in a natural setting: "He plays with his younger brother at the river." The zero-shot with UMR text is again unrelated, indicating a person performing an action in an unspecified setting: "The person is working there today." Then, the five-shot text contains more semantic similarity with the reference, as there is a male moving in the forest: "He has already started walking in the forest." Finally, the five-shot text with UMR shows the male to be running in the forest, clearly exhibiting the most semantic similarity with the reference: "He has already started running in the forest." This example exemplifies the fact that five-shot prompting

alone is not enough to achieve optimal performance on this task, and leveraging UMR in the prompt benefits performance beyond what is possible only when using demonstrations.

Another example is the Arápaho sentence, *wohei noh ci'ceese' hoo3itoo, heetnoo3itoone3en*, which has an English reference of "Wohei another story, I'm going to tell you another story." The zero-shot text is completely unrelated to the reference: "I walked to the store and said hello to the shopkeeper." The zero-shot with UMR output, on the other hand, is much more semantically similar to the reference: "I will tell you a story." The five-shot text then contains some of the specific language included in the reference, as well as some semantic similarity: "Wohei, then go ask him, the storyteller." Finally, the five-shot with UMR text is the most semantically similar, though the ending is slightly harder to interpret: "Wohei I will tell you a story about a little."

Our automatic metrics and qualitative analyses reveal that, on our test data, for most cases, incorporation of UMR graphs and demonstrations into the prompts enables heightened similarity with the reference English translation. Therefore, leveraging UMR in the prompt does indeed lead to heightened performance when translating from indigenous languages into English, while the greatest improvements are achieved by using related sentences as demonstrations; the combination of using both the demonstrations and the UMR in the prompt leads to the highest quality output in our experiments.

5 Conclusion

In this work, we begin to address one of the failings of GPT-based models: that of translation from extremely low-resource languages. We specifically examine the ability of a newly released Uniform Meaning Representation (UMR) dataset—containing sentences in Navajo, Arápaho, and Kukama, their UMR graphs, and their parallel sentences in English—to improve GPT-4 performance when included in the prompt. We find that both the incorporation of UMR graphs of the source text and adaptively selected demonstrations lead to improved performance on low-resource machine translation via prompting, with a statistically significant increase resulting in the majority of our comparisons. This is a promising indication of the downstream utility of UMR for low-resource settings and a step forward towards effective translation from indigenous languages via prompting.

Limitations

We perform experimentation on three extremely low-resource indigenous languages. Future work could expand this evaluation to other languages as well, varying in their depth of resources, as additional UMR annotations are released. UMR annotation can be expensive and time-consuming, as it requires fluency in the language and annotator training, which is a barrier to seamlessly incorporating UMR into downstream applications.

Additionally, we perform translation in one direction, with the low-resource languages serving as the source and English serving as the target language. Performing translation from English into these low-resource languages would make for interesting future work, though it will require a human evaluation by speakers of the language.

Finally, randomness is inherent in the results generated from GPT-4. We attempt to curtail this effect by providing statistical analyses for our findings, but further rigor could be added by running these experiments additional times.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Julia Bonn, Matthew J. Buchholz, Jayeol Chun, Andrew Cowell, William Croft, Lukas Denk, Sijia Ge, Jan Hajič, Kenneth Lai, James H. Martin, Skatje Myers, Alexis Palmer, Martha Palmer, Claire Benet Post, James Pustejovsky, Kristine Stenzel, Haibo Sun, Zdeňka Urešová, Rosa Vallejos, Jens E. L. Van Gysel, Meagan Vigus, Nianwen Xue, and Jin Zhao. 2024. Building a broad infrastructure for uniform meaning representations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2537–2547, Torino, Italy. ELRA and ICCL.

Allyson Ettinger, Jena Hwang, Valentina Pyatkin, Chandra Bhagavatula, and Yejin Choi. 2023. "You are an expert linguistic annotator": Limits of LLMs as analyzers of Abstract Meaning Representation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8250–8263, Singapore. Association for Computational Linguistics.

Sreyan Ghosh, Utkarsh Tyagi, Sonal Kumar, Chandra Kiran Evuru, Ramaneswaran S, S Sakshi, and Dinesh Manocha. 2024. ABEX: Data augmentation for low-resource NLU via expanding abstract descriptions. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 726–748, Bangkok, Thailand. Association for Computational Linguistics.

Ping Guo, Yubing Ren, Yue Hu, Yunpeng Li, Jiarui Zhang, Xingsheng Zhang, and Heyan Huang. 2024. Teaching large language models to translate on low-resource languages with textbook prompting. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15685–15697, Torino, Italia. ELRA and ICCL.

Sireesh Gururaja, Ritam Dutt, Tinglong Liao, and Carolyn Rosé. 2023. Linguistic representations for fewer-shot relation extraction across domains. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7502–7514, Toronto, Canada. Association for Computational Linguistics.

Yilun Hua, Zhaoyuan Deng, and Kathleen McKeown. 2023. Improving long dialogue summarization with semantic graph representation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13851–13883, Toronto, Canada. Association for Computational Linguistics.

Robert T. Kasper. 1989. A flexible interface for linking applications to Penman's sentence generator. In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989*.

Wai Ching Leung, Shira Wein, and Nathan Schneider. 2022. Semantic similarity as a window into vector- and graph-based metrics. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 106–115, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Open AI. 2022. Introducing ChatGPT.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards making the most of ChatGPT for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5622–5633, Singapore. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Nathaniel Robinson, Perez Ogayo, David R. Mortensen, and Graham Neubig. 2023. ChatGPT MT: Competitive for high- (but not low-) resource languages. In *Proceedings of the Eighth Conference on Machine Translation*, pages 392–418, Singapore. Association for Computational Linguistics.

David Stap and Ali Araabi. 2023. ChatGPT is not a good indigenous translator. In *Proceedings of the Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP)*, pages 163–167, Toronto, Canada. Association for Computational Linguistics.

Lei Tang, Jinghui Qin, Wenxuan Ye, Hao Tan, and Zhijing Yang. 2025. Adaptive few-shot prompting for machine translation with pre-trained language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(24):25255–25263.

Jens E. L. Van Gysel, Meagan Vigus, Lukas Denk, Andrew Cowell, Rosa Vallejos, Tim O'Gorman, and William Croft. 2021a. Theoretical and practical issues in the semantic annotation of four indigenous languages. In *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 12–22, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jens E. L. Van Gysel, Meagan Vigus, Pavlina Kalm, Sook-kyung Lee, Michael Regan, and William Croft. 2019. Cross-linguistic semantic annotation: Reconciling the language-specific and the universal. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 1–14, Florence, Italy. Association for Computational Linguistics.

Jens E.L. Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. 2021b. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3):343–360.

Meagan Vigus, Jens E. L. Van Gysel, Tim O'Gorman, Andrew Cowell, Rosa Vallejos, and William Croft. 2020. Cross-lingual annotation: a road map for low- and no-resource languages. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 30–40, Barcelona Spain (online). Association for Computational Linguistics.

Shira Wein and Julia Bonn. 2023. Comparing UMR and cross-lingual adaptations of AMR. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations*, pages 23–33, Nancy, France. Association for Computational Linguistics.

Shira Wein and Juri Opitz. 2024. A survey of AMR applications. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6856–6875, Miami, Florida, USA. Association for Computational Linguistics.

Shira Wein and Nathan Schneider. 2024. Assessing the cross-linguistic utility of Abstract Meaning Representation. *Computational Linguistics*, 50(2):419–473.

Shira Wein, Zhuxin Wang, and Nathan Schneider. 2023. Measuring fine-grained semantic equivalence with Abstract Meaning Representation. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 144–154, Nancy, France. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *Proc. of ICLR*, Online.

A Prompting Protocols

Zero-shot

```
Please provide the English translation for this [Source language] sentence. Do not provide
any explanations or text apart from the translation.
[Source Language]: [sentence to be translated]
English:
```

Zero-shot with UMR

```
Please provide the English translation for this [Source language] sentence (which is
accompanied by a Uniform Meaning Representation parse). Do not provide any explanations or text apart
from the translation.
[Source language]: [sentence to be translated]
Uniform Meaning Representation: [UMR of source text]
English:
```

Five-shot

```
Please provide the English translation for this [Source language] sentence. Do not provide
any explanations or text apart from the translation.
[Source language]: [sentence 1] English: [translation 1]
[Source language]: [sentence 2] English: [translation 2]
[Source language]: [sentence 3] English: [translation 3]
[Source language]: [sentence 4] English: [translation 4]
[Source language]: [sentence 5] English: [translation 5]
Please provide the English translation for this [Source language] sentence.
Do not provide any explanations or text apart from the translation.
[Source language]: [sentence to be translated]
English:
```

Five-shot with UMR

```
Please provide the English translation for this [Source language] sentence (which is
accompanied by a Uniform Meaning Representation parse). Do not provide any explanations or text apart
from the translation.
[Source language]: [sentence 1] Uniform Meaning Representation: [UMR 1] English: [translation 1]
[Source language]: [sentence 2] Uniform Meaning Representation: [UMR 2] English: [translation 2]
[Source language]: [sentence 3] Uniform Meaning Representation: [UMR 3] English: [translation 3]
[Source language]: [sentence 4] Uniform Meaning Representation: [UMR 4] English: [translation 4]
[Source language]: [sentence 5] Uniform Meaning Representation: [UMR 5] English: [translation 5]
Please provide the English translation for this [Source language] sentence (which is accompanied by
a Uniform Meaning Representation parse). Do not provide any explanations or text apart from the
translation.
[Source language]: [sentence to be translated]
Uniform Meaning Representation: [UMR of source text]
English:
```

Figure 2: The "user" portions of our prompts for the four protocols. For all prompts, the "system" portion of the protocol is as follows: `System: You are a machine translation system from [Source language] to English that translates sentences from narrative documents.`

Subword models struggle with word learning, but surprisal hides it

Bastian Bunzeck and **Sina Zarrieß**
Computational Linguistics, Department of Linguistics
Bielefeld University, Germany
{bastian.bunzeck, sina.zarriess}@uni-bielefeld.de

Abstract

We study word learning in subword and character language models with the psycholinguistic lexical decision task. While subword LMs struggle to discern words and non-words with high accuracy, character LMs solve this task easily and consistently. Only when supplied with further contexts do subword LMs perform similarly to character models. Additionally, when looking at word-level and syntactic learning trajectories, we find that both processes are separable in character LMs. Word learning happens before syntactic learning, whereas both occur simultaneously in subword LMs. This raises questions about the adequacy of subword LMs for modeling language acquisition and positions character LMs as a viable alternative to study processes below the syntactic level.

1 Introduction

When humans acquire their first language(s), they *first* learn to recognize single words, mostly from short, fragmentary utterances (Cameron-Faulkner et al., 2003; Bunzeck and Diessel, 2024), *before* fully understanding the grammatical processes governing them (Tomasello, 1992; Behrens, 2021). This simple fact about language acquisition has received surprisingly little attention in the body of work that treats LMs as models of language learners (Warstadt and Bowman, 2022; Portelance and Jasbi, 2024). While word learning in children is comparatively well studied (Plunkett, 1997; Yu and Ballard, 2007; Waxman and Gelman, 2009; Bergelson and Swingley, 2012; Clark and Casillas, 2015; Frank et al., 2021), the implicit word learning processes in LMs are not. Current studies focus on syntax (Mueller et al., 2022; Choshen et al., 2022), or investigate word learning in close connection with syntax through surprisal (Chang and Bergen, 2022; Portelance et al., 2023; Shafiabadi and Wisniewski, 2025; Ficarra et al., 2025). Architecture-wise, a key limitation to the precise study of word learning is subword tokenization (e.g., BPE, Gage, 1994), which splits words into linguistically (Arnett and Bergen, 2025) and cognitively implausible units (Beinborn and Pinter, 2023).

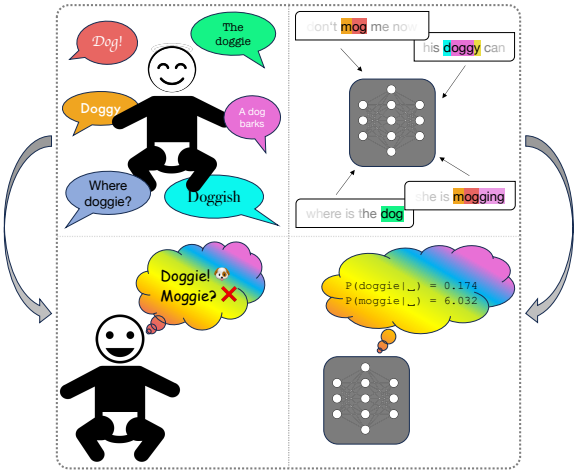


Figure 1: Illustration of word learning in human learners and transformer LLMs (top), and of our lexical decision test that probes discrimination of words from non-words (bottom). While human learners build up an mental lexicon from experience with language, artificial learners assign probabilities to strings based on their frequency.

To gauge word learning in a syntax-independent manner, we use the psycholinguistic lexical decision task (Meyer and Schvaneveldt, 1971; Le Godais et al., 2017), i.e., deciding which word in a given word/non-word pair is real. We find that models with character-level tokenization learn this task quickly and reliably. In contrast, subword LMs of all sizes perform significantly worse in a syntax-independent setting and only achieve comparable accuracy when stimuli are measured through surprisal, or "unexpectedness," in linguistic context. By comparing word and syntactic learning (measured via BLiMP, Warstadt et al., 2020), we further find that character models quickly acquire word knowledge and only later develop syntactic knowledge. In subword models, word and syntax learning happen concurrently. This demonstrates how ele-

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 286–300
July 27 - August 1, 2025

mentary modeling decisions, such as tokenization methods, significantly impact learning trajectories in LMs, a fact that warrants more scrutiny when using LMs as models of language acquisition.

2 Related work

Word learning in humans is a multifaceted phenomenon that involves different kinds of (extra)linguistic knowledge (Waxman and Gelman, 2009): while phoneticians are concerned with word recognition and sequence segmentation (Jusczyk, 1999), developmental psychologists frequently equate word learning with correct reference to real world objects (Stager and Werker, 1997; Ackermann et al., 2020). Psycholinguists usually focus on the mental lexicon and learning which words belong to it (Goldinger, 1996), whereas usage-based scholars take into account children's productions and their ability to be competent language users, even with few words (Tomasello, 1992).

Although aspects like word recognition and sequence segmentation have been studied in LMs (e.g. Goriely and Buttery, 2025a), the most common approach to word learning in LMs is measuring the predictability of words via surprisal (negative log-probability, Hale, 2001). Chang and Bergen (2022) train LMs on book texts and wiki data. They define a surprisal threshold below which words are said to be learned and find that frequent function words are learned earliest. Here, the alignment between models and real learners is questionable: children first utter nouns and verbs (Tomasello, 2000), but also rely on function words for challenges like speech segmentation (Dye et al., 2019). Portelance et al. (2023) show that in LSTMs trained on child-directed speech, surprisal correlates with word-level age of acquisition. Chang et al. (2024) observe that learning curves for surprisal values are stable for frequent tokens, while infrequent tokens are "forgotten" again over pre-training. Shafiabadi and Wisniewski (2025) introduce anti-surprisal (in incorrect contexts) to track false usage, which also fluctuates over pre-training. These studies cast word learning as the ability to anticipate words' expectedness in a given syntactic (and semantic) context. We note a certain conceptual leap to the original works on surprisal, where it is primarily viewed as an incremental measure of processing difficulty in syntactic comprehension (Levy, 2008; Demberg and Keller, 2009). A simple word like *dog* might be surprising and therefore hard to parse in some contexts, but very expected in others, independently of being already learned on the pure word level. A further methodological drawback of surprisal as a measure of word learning is that it corresponds almost directly to the next-token prediction objective LMs are trained on. This contrasts with typical probing paradigms used in the domain of syntax, which implement the idea to "challenge" models in minimal pair set-ups that are not observed directly as string sequences in training, thereby testing abstracted, implicit linguistic knowledge rather than observed patterns in the data. In a similar vein, we want to probe the word knowledge of an LM at a fundamental level and beyond surface-level word sequences that LMs are known to excel in predicting. We want to know if the artificial learner knows that the word *doggie* exists in the English language, but *moggie* does not.

Lexical decision is widely used in human studies but remains an underexplored LM benchmark. Le Godais et al. (2017) show that character-based LSTMs achieve about 95% accuracy on such tasks. Lavechin et al. (2023) find that speech-based LMs need significantly more input to still perform poorly (56.8%) than phoneme-level LSTMs (75.4%) on a phonetic dataset. For the same data, Goriely et al. (2024) find that GPT-2-based subword BabyLMs achieve 70% accuracy and a comparable character models reach nearly 90%. Finally, for another lexical decision dataset, Bunzeck et al. (2025) report near-perfect accuracy for character-based grapheme Llama models, while phoneme models perform at 60–70%.

3 Experiments

Models We train triplets of increasingly larger Llama models (Touvron et al., 2023) with character/subword tokenization on the BabyLM 10M corpus (Choshen et al., 2024). Training details are found in Appendix A. As ablations, we test subword Pythias (Biderman et al., 2023) and character/subword GPT-2 models (Goriely et al., 2024).

Test data We follow the idea of forced-choice lexical decision (Baddeley et al., 1993), where participants must decide which is real: an existing word or a synthesized non-word. We use `wuggy` (Keuleers and Brysbaert, 2010) to generate minimal pairs of words/non-words that differ in one or two syllables, akin to syntactic minimal pair tests such as BLiMP. We derive 1,000 non-words (e.g. *monding*) each from 1,000 high-frequency/low-

Tokenization	Model	Parameters	Data size	Lexical decision highFrq	Lexical decision lowFrq	Surprisal highFrq	Surprisal lowFrq	Anti-surprisal highFrq	Anti-surprisal lowFrq
Subword (BPE)	Pythia	14M	825GB	66.6	62.5	90.5	85.5	71.4	77.7
		70M		72.5	68.8	94.5	94.0	77.0	83.6
		160M		77.8	73.0	96.4	95.8	78.0	85.7
		410M		81.9	78.1	97.7	97.9	77.1	84.1
		1B		87.5	83.2	97.7	97.9	76.6	83.8
		1.4B		87.8	81.6	97.9	97.9	76.5	84.7
	GPT-2	97.5M	100M words	35.6	79.1	99.0	99.2	84.7	86.9
	Llama	2.51M	10M words	70.9	58.4	86.7	70.9	78.6	67.7
		7.77M		79.5	63.2	91.3	78.1	81.1	72.9
		30.03M		83.6	68.6	92.7	81.1	83.7	76.1
Character	GPT-2	85.3M	100M words	98.7	**97.3**	**99.8**	**99.4**	98.0	**96.3**
	Llama	0.49M	10M words	97.6	83.0	98.2	84.3	98.0	83.1
		3.73M		98.9	90.2	99.4	90.3	98.5	88.8
		21.94M		**99.0**	93.3	**99.8**	94.7	**99.0**	92.5

Table 1: Accuracy scores (in %) for (i) lexical decision, (ii) surprisal and (iii) anti-surprisal experiments

frequency words (e.g. *sending*), which preserve syllable-bigram frequencies and match their origin words in length (cf. Appendix B).

Lexical decision For a word/non-word pair $(w, *w)$, we measure $-log(P(w|\_))$ and $-log(P(*w|\_))$, i.e. how "surprised" a LM is by the word in the context of a prepended whitespace (and BOS token). If $-log(P(w|\_)) < -log(P(*w|\_))$, the LM's lexical decision is correct. As autoregressive LMs are sequence prediction models, we need a preceding context for which we can calculate surprisal. A single whitespace is the most neutral starting token available (and for subword models also signals that the first subword is word-initial). For all experiments, we calculate the average surprisal over all tokens of a word (which, in some cases, is characterized by a mismatch in token numbers between words and non-words, cf. Appendix B) with `minicons` (Misra, 2022).

Surprisal To measure LMs' knowledge of words presented in regular syntactic contexts, we calculate the surprisal of words and non-words $(w, *w)$ as $-log(P(w_i|w_{n<i}))$, i.e. the degree to which the LM is "surprised" by the word in the context of plausible preceding tokens, including a BOS token. We create stimuli by sampling sentences that contain our target words from OpenSubtitles (Lison and Tiedemann, 2016) and substituting them with matching non-words for the false stimuli. If $-log(P(w_i|w_{n<i})) < -log(P(*w_i|w_{n<i}))$, the LM's decision is correct.

Anti-surprisal Inspired by Shafiabadi and Wisniewski (2025), we include anti-surprisal, a measure of word surprisal on negative instances. We create negative samples by selecting sentences that our original words do not occur in, and then randomly[1] placing words/non-words into these sentences at the same index, where $index \geq 3$. By doing so, we compromise between lexical decision and surprisal measurement. There are two reasons to include this measure: i) surprisal in negative samples provides the model with word material as context, but without semantic or syntactic signals that could prime the model towards recognizing it; this allows us to assess whether the mere presence of other words in context makes it easier for the model to distinguish words from non-words, compared to our lexical decision set-up where only whitespace is given. In addition, ii) we want to see if the presence of an ill-fitting context actively deteriorates performance in the sense of a model suddenly preferring non-words over existing words. Again, if $-log(P(w_i|w_{n<i})) < -log(P(*w_i|w_{n<i}))$, the LM's decision is correct.

Learning trajectories To assess when word learning happens in relation to syntax learning, we further evaluate intermediate checkpoints of our models on our word learning tests and BLiMP as a syntactic benchmark. In line with previous studies (Chang and Bergen, 2022; Viering and Loog, 2023), we space our checkpoints logarithmically – 10 for the first 10% of training, 9 more for the remaining 90%. For the Pythia models, we extract similarly spaced checkpoints (the GPT-2 models do not provide checkpoints, so we exclude them).

[1] In line with Shafiabadi and Wisniewski (2025), we do not further characterize the resulting violations on a syntactic or semantic level.

4 Results

Lexical decision The lexical decision results (Table 1) show a strong contrast between character and subword models. Character models achieve near-perfect accuracy (97–99%) on high-frequency words, regardless of model size. The performance on low-frequency words steadily increases with model size and reaches a near-perfect level for our largest character Llama and the character GPT-2. On the other hand, all BPE models get surprisingly low scores on high-frequency words: The smallest Pythia model discriminates between word and non-words with an accuracy of only 67%, and the BPE GPT-2 performs below the chance baseline. Even the largest and best BPE model reaches only 87.8% on high-frequency words – almost 10% less than the smallest character model.

Scaling laws generally hold, with larger models outperforming smaller ones. Interestingly, for the BPE models, there is a consistent gap between high and low-frequency words that cannot be closed by larger models. Smaller character models also show a performance gap between high and low-frequency words, but it narrows considerably with larger models. These results point to substantial differences in how subword and character models learn words. Such a surprising lack of ability in distinguishing words from non-words (without context) is a blatant, hitherto overlooked gap in subword models.

Surprisal and anti-surprisal Results for the second experiment (Table 1) differ from those for lexical decision. In the surprisal setting, the difference between BPE and character models is less pronounced. On high-frequency data, nearly all models (except the smallest BPE Llama) achieve over 90% accuracy. Still, larger character models yield the best results. In the low-frequency data condition, the pattern is similar, though scores are generally lower. Very large BPE models outperform our Llamas there, but the character GPT-2 remains superior. This may be attributed to the limited lexical exposure of our Llamas, trained on only 10M tokens. In the anti-surprisal setting, character models again drastically outperform BPE models and achieve nearly perfect scores on high-frequency data, while BPE models only reach 70–80% accuracy. This setting is the only one where Pythia models get better scores on low-frequency data, with a gap of 6–8%, which increases with model size (not the case for BPE Llamas or character models). This contrast between surprisal and antisuprisal might

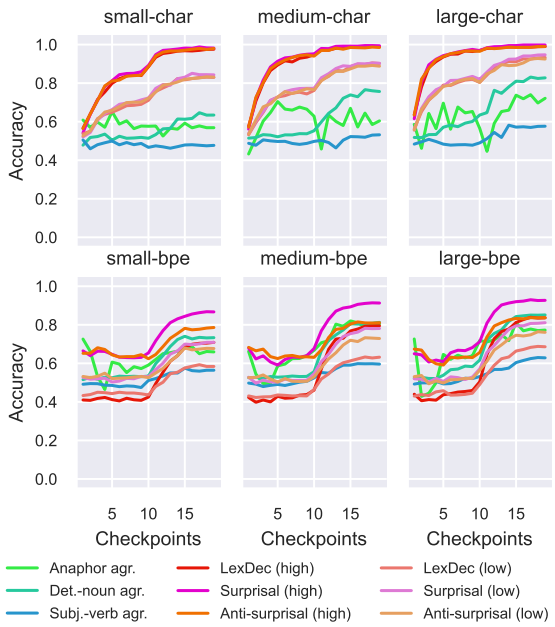


Figure 2: Selected lexical and syntactic learning curves

be an indicator of the entanglement of word learning and syntactic learning in subword models. It is plausible that for high-frequency words, the BPE models have strong expectations about which word should come next in a certain context, and because this expectation is not matched by the real (but ill-fitting) word, a made-up non-word is preferred. We argue that this should still not be the case for an ideal language model – if a model is indeed well-tuned, it should assign a higher probability to an ill-fitting but existing word which is still in distribution, than to a completely ill-fitting string that is out-of-distribution. In any case, BPE models catch up to character models if (and *only* if) provided with additional syntactic/semantic context information. While random context somewhat aids BPE models, a substantial gap remains between the largest BPE models and the character models, where performance remains excellent, even in the presence of implausible contexts.

Learning trajectories Figure 2 displays learning curves for syntactic agreement phenomena, lexical decision, and both surprisal conditions across the 19 saved checkpoints (complete curves reproduced in Appendix C). The first 10 checkpoints correspond to the first 10% of pretraining, the remaining 9 checkpoints represent 10% of training steps over the remaining 90% of pretraining. For character models, the high-frequency, low-frequency, and syntactic curves are clearly separated. On high-frequency data, word learning is rapid and follows power-law curves; the low-frequency scores im-

prove a little later and at a lower rate, but with the same trajectory. Syntactic phenomena improve later, mostly in s-shaped curves (e.g., det.-noun agreement), or are not learned at all in small models (subj.-verb agreement). In contrast, the syntactic and lexical curves for BPE models form sheaves of s-shaped trajectories. There is *no* principal difference in learning dynamics between the syntactic and the word level, improvements[2] occur simultaneously. This further confirms the results of our previous experiments: in BPE models, word learning is dependent on syntax learning, and words cannot be recognized reliably outside of plausible contexts. Additionally, these different levels of learning cannot be disentangled in BPE models, whereas in character models, syntactic learning *follows* word learning.[3]

5 Discussion

In contrast to previous work, this study set out to explore when (and if) LMs learn what valid words *are*, and not how LMs learn when words are validly *used*. How should these results now be interpreted in the light of language acquisition and the use of BabyLMs to model corresponding processes? In reality, words are not learned in isolation, but from usage. Yet, words have been widely shown to be represented as solid "standalone" units in the mental lexicon, and to be units that human learners acquire early on (cf. Waxman and Gelman, 2009, also Montag et al., 2018). Usage-based approaches have tested aspects of word learning independently from syntax (mostly in object naming tasks, cf. Tomasello and Todd, 1983) and relate it to pragmatic aspects of communication. For example, children only react to words they know and ignore similar-sounding words. They even struggle with learning words that are phonetically extremely similar (like our stimuli), and only later gain this capacity (Stager and Werker, 1997). Similarly, in production, the earliest words come in isolation, slowly emerge into pivot schemas and holophrases (Tomasello, 1992, 2003), and only then finally turn into complex sentences. Of course, syntax also aids in discovering aspects of "wordiness", like SV(X)-sentences offering cues for agent-patient relationships. It would be very interesting to further disentangle these levels of word knowledge in a follow-up study, but our current study is not concerned with the "you shall know a word by the company it keeps"-level of word knowledge, but rather with the "what do valid words of the language look like"-level, which can be assessed via lexical decision. As such, we believe that the separated learning curves of our character-level models represent more human-like learning than the highly correlated curves found in the subword models, but in reality, an overlap between them is definitely expected.

Reasons for the tremendous performance differences between subword and character models remain open to further inquiry. One plausible explanation is that character models have much more context available to calculate meaningful sequence probabilities, as words are split into many more tokens. While this is true, it is also exactly the point that we are stressing here: it is hard to imagine that arbitrary subword units lead to human-like, plausible word-level representations (like in exemplar models of lexical storage, cf. Bybee, 2010), whereas character models might offer a more justified level of granularity (and, e.g., better fit reading times, cf. Oh et al., 2021). Our findings also align with results on LMs' sensitivity to character-level perturbations (Moradi and Samwald, 2021; Zhu et al., 2024) and their inability to solve character-level tasks, like counting occurrences of the letter *r* in the word *strawberry* (Zhang and He, 2024; Shin and Kaneko, 2024; Cosma et al., 2025).

6 Conclusion

We have shown that the lexical decision approach to the study of word learning in LMs complements surprisal-based approaches and reveals difficulties that surprisal hides: subword LMs struggle with lexical decision, whereas character models master this task with ease. Additionally, in subword LMs, lexical and syntactic learning are inseparable, whereas word learning in character models precedes syntactic learning; the processes are related, yet separable. It is plausible that the *a priori* token splitting in subword models preempts a word discovery process in them, whereas character models first have to pass through this developmental stage, possibly in a somewhat more human-like manner. In any case, as we have shown, decisions about the representational levels of LMs tremendously influence their learning pathways on the different levels of linguistic analysis.

[2] We calculate Spearman-rank correlation between ordered accuracies for word-level tasks and BLiMP in Appendix D.

[3] Interestingly, the learning process is not finished when accuracy curves stabilize, cf. Appendix F.

Limitations

The generalizability of our findings is constrained by a few factors. The present study has only focused on the English language, but it is plausible that other languages with different writing systems or graphematic and phonotactic rule systems exhibit different patterns under different tokenization schemes. Here, phonetic transcriptions might provide a viable alternative, but real narrow transcriptions that accurately capture the whole breadth of human input are scarce and extremely costly and laborious to manually produce (although novel datasets like Goriely and Buttery, 2025b provide an alternative through automatic transcription). Besides, for the character LMs, we focus only on small models, as very large models with such tokenization, especially ones providing intermediate checkpoints, are nonexistent at this moment. It would still be interesting to see how they compare to subword models in a setting where parameter size and training data are greatly increased.

As already mentioned in Section 2, from a developmental perspective, word learning in humans also includes other processes than statistical pattern recognition from the input: semantic aspects and real-world reference are equally important, as are multimodal input and communicative intent. The ongoing form-vs.-function debate on LMs (Mahowald et al., 2024) has begun to consider these aspects, and further studies should aim at incorporating them; for example, the object naming paradigm used in many developmental psychology studies would lend itself naturally to the study of multimodal models.

Finally, we also want to mention that there are attempts to add more linguistic theory to tokenizers. Looking into different tokenizers that, for example, try to be more morphology-aware (Hofmann et al., 2022; Bauwens and Delobelle, 2024; Yehezkel and Pinter, 2023; Uzan et al., 2024) or implement other optimization tricks (Schmidt et al., 2024) could yield even more fine-grained points of comparison, but for the present study and its limited scope we focused on the most popular tokenization scheme (BPE) and the most linguistically minimalist alternative (characters).

Ethical considerations

Due to the nature of this work, no concrete ethical aspects or repercussions need to be discussed. However, we would like to stress that, of course, BabyLMs not supposed to simulate real babies, but only abstractions of a very specific part of their learning capacity (frequency-driven, domain-general learning mechanisms such as entrenchment or resonance), and therefore all claims about their implications for language development in the real world should be interpreted in this light.

Acknowledgments

We would like to thank the three anonymous ARR reviewers for their insightful comments and the highly valuable discussions throughout the rebuttal period.

This research has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – CRC-1646, project number 512393437, project A02.

References

Lena Ackermann, Robert Hepach, and Nivedita Mani. 2020. Children learn words easier when they are interested in the category to which the word belongs. *Developmental Science*, 23(3):e12915.

Philip A. Allen, Albert F. Smith, Mei-Ching Lien, Jeremy Grabbe, and Martin D. Murphy. 2005. Evidence for an Activation Locus of the Word-Frequency Effect in Lexical Decision. *Journal of Experimental Psychology: Human Perception and Performance*, 31(4):713–721.

Catherine Arnett and Benjamin Bergen. 2025. Why do language models perform worse for morphologically complex languages? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6607–6623, Abu Dhabi, UAE. Association for Computational Linguistics.

Harald Baayen, Piepenbrock R, and Gulikers L. 1995. CELEX2.

Alan Baddeley, Hazel Emslie, and Ian Nimmo-Smith. 1993. The Spot-the-Word test: A robust estimate of verbal intelligence based on lexical decision. *British Journal of Clinical Psychology*, 32(1):55–65.

Thomas Bauwens and Pieter Delobelle. 2024. BPE-knockout: Pruning Pre-existing BPE Tokenisers with Backwards-compatible Morphological Semi-supervision. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5810–5832, Mexico City, Mexico. Association for Computational Linguistics.

Heike Behrens. 2021. Constructivist Approaches to First Language Acquisition. *Journal of Child Language*, 48(5):959–983.

Lisa Beinborn and Yuval Pinter. 2023. Analyzing Cognitive Plausibility of Subword Tokenization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4478–4486, Singapore. Association for Computational Linguistics.

Elika Bergelson and Daniel Swingley. 2012. At 6–9 months, human infants know the meanings of many common nouns. *Proceedings of the National Academy of Sciences*, 109(9):3253–3258.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. *Preprint*, arXiv:2304.01373.

Bastian Bunzeck and Holger Diessel. 2024. The richness of the stimulus: Constructional variation and development in child-directed speech. *First Language*, 45(2):152–176.

Bastian Bunzeck, Daniel Duran, Leonie Schade, and Sina Zarrieß. 2025. Small language models also work with small vocabularies: Probing the linguistic abilities of grapheme- and phoneme-based baby llamas. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6039–6048, Abu Dhabi, UAE. Association for Computational Linguistics.

Joan Bybee. 2010. *Language, Usage and Cognition*, 1 edition. Cambridge University Press.

Thea Cameron-Faulkner, Elena Lieven, and Michael Tomasello. 2003. A construction based analysis of child directed speech. *Cognitive Science*, 27(6):843–873.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word Acquisition in Neural Language Models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Tyler A. Chang, Zhuowen Tu, and Benjamin K. Bergen. 2024. Characterizing Learning Curves During Language Model Pre-Training: Learning, Forgetting, and Stability. *Transactions of the Association for Computational Linguistics*, 12:1346–1362.

Leshem Choshen, Ryan Cotterell, Michael Y. Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. [Call for Papers] The 2nd BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Preprint*, arXiv:2404.06214.

Leshem Choshen, Guy Hacohen, Daphna Weinshall, and Omri Abend. 2022. The Grammar-Learning Trajectories of Neural Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8281–8297, Dublin, Ireland. Association for Computational Linguistics.

Eve V Clark and Marisa Casillas. 2015. First language acquisition. In *The Routledge Handbook of Linguistics*, pages 311–328. Routledge.

Adrian Cosma, Stefan Ruseti, Emilian Radoi, and Mihai Dascalu. 2025. The Strawberry Problem: Emergence of Character-level Understanding in Tokenized Language Models. *arXiv preprint*.

Vera Demberg and Frank Keller. 2009. A Computational Model of Prediction in Human Parsing: Unifying Locality and Surprisal Effects. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 31, pages 1888–1893.

Cristina Dye, Yarden Kedar, and Barbara Lust. 2019. From lexical to functional categories: New foundations for the study of language development. *First Language*, 39(1):9–32.

Filippo Ficarra, Ryan Cotterell, and Alex Warstadt. 2025. A Distributional Perspective on Word Learning in Neural Language Models. *arXiv preprint*.

Michael C. Frank, Mika Braginsky, Daniel Yurovsky, and Virginia A. Marchman. 2021. *Variability and Consistency in Early Language Learning: The Wordbank Project*. The MIT Press.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *Preprint*, arXiv:2101.00027.

Stephen D. Goldinger. 1996. Auditory Lexical Decision. *Language and Cognitive Processes*, 11(6):559–568.

Zébulon Goriely and Paula Buttery. 2025a. BabyLM's First Words: Word Segmentation as a Phonological Probing Task. *arXiv preprint*.

Zébulon Goriely and Paula Buttery. 2025b. IPA-CHILDES & G2P+: Feature-Rich Resources for Cross-Lingual Phonology and Phonemic Language Modeling. *arXiv preprint*.

Zébulon Goriely, Richard Diehl Martinez, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2024. From babble to words: Pre-training language models on continuous streams of phonemes. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 37–53, Miami, FL, USA. Association for Computational Linguistics.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Valentin Hofmann, Hinrich Schuetze, and Janet Pierrehumbert. 2022. An Embarrassingly Simple Method to Mitigate Undesirable Properties of Pretrained Language Model Tokenizers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–393, Dublin, Ireland. Association for Computational Linguistics.

Peter W. Jusczyk. 1999. How infants begin to extract words from speech. *Trends in Cognitive Sciences*, 3(9):323–328.

Emmanuel Keuleers and Marc Brysbaert. 2010. Wuggy: A multilingual pseudoword generator. *Behavior Research Methods*, 42(3):627–633.

Marvin Lavechin, Yaya Sy, Hadrien Titeux, María Andrea Cruz Blandón, Okko Räsänen, Hervé Bredin, Emmanuel Dupoux, and Alejandrina Cristia. 2023. BabySLM: Language-acquisition-friendly benchmark of self-supervised spoken language models. In *INTERSPEECH 2023*, pages 4588–4592. ISCA.

Gaël Le Godais, Tal Linzen, and Emmanuel Dupoux. 2017. Comparing character-level neural language models using a lexical decision task. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 125–130, Valencia, Spain. Association for Computational Linguistics.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2024. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*, pages 517–540.

James L. McClelland and David E. Rumelhart. 1981. An interactive activation model of context effects in letter perception: I. An account of basic findings. *Psychological Review*, 88(5):375–407.

David E. Meyer and Roger W. Schvaneveldt. 1971. Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology*, 90(2):227–234.

Kanishka Misra. 2022. Minicons: Enabling Flexible Behavioral and Representational Analyses of Transformer Language Models. *arXiv preprint*.

Jessica L. Montag, Michael N. Jones, and Linda B. Smith. 2018. Quantity and Diversity: Simulating Early Word Learning Environments. *Cognitive Science*, 42(S2):375–412.

Milad Moradi and Matthias Samwald. 2021. Evaluating the Robustness of Neural Language Models to Input Perturbations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1558–1570, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aaron Mueller, Robert Frank, Tal Linzen, Luheng Wang, and Sebastian Schuster. 2022. Coloring the Blank Slate: Pre-training Imparts a Hierarchical Inductive Bias to Sequence-to-sequence Models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1352–1368, Dublin, Ireland. Association for Computational Linguistics.

Byung-Doh Oh, Christian Clark, and William Schuler. 2021. Surprisal Estimators for Human Reading Times Need Character Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3746–3757, Online. Association for Computational Linguistics.

Kim Plunkett. 1997. Theories of early language acquisition. *Trends in Cognitive Sciences*, 1(4):146–153.

Eva Portelance, Yuguang Duan, Michael C. Frank, and Gary Lupyan. 2023. Predicting Age of Acquisition for Children's Early Vocabulary in Five Languages Using Language Model Surprisal. *Cognitive Science*, 47(9):e13334.

Eva Portelance and Masoud Jasbi. 2024. The Roles of Neural Networks in Language Acquisition. *Language and Linguistics Compass*, 18(6):e70001.

Craig W Schmidt, Varshini Reddy, Haoran Zhang, Alec Alameddine, Omri Uzan, Yuval Pinter, and Chris Tanner. 2024. Tokenization Is More Than Compression. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 678–702, Miami, Florida, USA. Association for Computational Linguistics.

Nazanin Shafiabadi and Guillaume Wisniewski. 2025. Beyond surprisal: A dual metric framework for lexical skill acquisition in LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6636–6641, Abu Dhabi, UAE. Association for Computational Linguistics.

Andrew Shin and Kunitake Kaneko. 2024. Large language models lack understanding of character composition of words. In *ICML 2024 Workshop on Llms and Cognition*.

Christine L. Stager and Janet F. Werker. 1997. Infants listen for more phonetic detail in speech perception than in word-learning tasks. *Nature*, 388(6640):381–382.

Michael Tomasello. 1992. *First Verbs: A Case Study of Early Grammatical Development*, 1 edition. Cambridge University Press.

Michael Tomasello. 2000. The item-based nature of children's early syntactic development. *Trends in Cognitive Sciences*, 4(4).

Michael Tomasello. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press.

Michael Tomasello and Jody Todd. 1983. Joint attention and lexical acquisition style. *First Language*, 4(12):197–211.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *Preprint*, arXiv:2302.13971.

Omri Uzan, Craig W. Schmidt, Chris Tanner, and Yuval Pinter. 2024. Greed is All You Need: An Evaluation of Tokenizer Inference Methods. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 813–822, Bangkok, Thailand. Association for Computational Linguistics.

Tom Viering and Marco Loog. 2023. The Shape of Learning Curves: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7799–7819.

Alex Warstadt and Samuel R. Bowman. 2022. *What Artificial Neural Networks Can Tell Us about Human Language Acquisition*, 1 edition, pages 17–60. CRC Press, Boca Raton.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Gotlieb Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–6, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Sandra R. Waxman and Susan A. Gelman. 2009. Early word-learning entails reference, not merely associations. *Trends in Cognitive Sciences*, 13(6):258–263.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick Von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shaked Yehezkel and Yuval Pinter. 2023. Incorporating context into subword vocabularies. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 623–635, Dubrovnik, Croatia. Association for Computational Linguistics.

Chen Yu and Dana H. Ballard. 2007. A unified model of early word learning: Integrating statistical and social cues. *Neurocomputing*, 70(13-15):2149–2165.

Yidan Zhang and Zhenan He. 2024. Large Language Models Can Not Perform Well in Understanding and Manipulating Natural Language at Both Character and Word Levels? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11826–11842, Miami, Florida, USA. Association for Computational Linguistics.

Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. 2024. PromptBench: A unified library for evaluation of large language models. *Journal of Machine Learning Research*, 25(254):1–22.

A Model hyperparameters and training details

We use the `transformers` library (Wolf et al., 2020) to train our models. The corresponding hyperparameters are listed in Table 2. We opted for very small LMs and little training data because the BabyLM paradigm has consolidated itself as a well-established method of developmentally plausible language modeling (Warstadt et al., 2023; Choshen et al., 2024). Also, we noticed the word learning to occur quite rapidly in our models, so we argue that small LMs offer more fine-grained opportunities for investigating these processes – in larger LMs, even a singular training step can already influence performance on such a brittle task like lexical decision tremendously.

The subword models feature a considerably higher number of parameters, as the embedding layer of transformer language models accounts for a quite large share of overall model parameters. In light of our vastly different vocabulary sizes (102 for character models, 8,002 for subword models), these differences are not surprising. The subword tokenizer is a regular BPE tokenizer self-trained with the `tokenizers` library and contains 8,000 subword tokens, a beginning-of-sequence token and a combined end-of-sequence/padding token. The character tokenizer contains these two special tokens and all printable ASCII characters, which are sufficient to represent all graphemes of the English language.

Our models were trained on a Apple M2 Pro processor with the MPS backend. For the small models, training lasted approx. 20min, for the medium-sized models it took approx. 1h and for the largest models approx. 10h (no principal differences between character and subword models). We share our models with their checkpoints on the HuggingFace hub.[4]

Loss curves for all models can be found in Figure 3. For the test loss, we calculated the perplexity over a held-out portion of our training corpus that is comparable in composition to the training data. We find no principal differences in loss development, although the character models converge faster. Larger models also tend to converge faster and generally reach smaller absolute loss values. As the similar train and test loss curves indicate, all Llama models succeed in optimizing for their next-token prediction objective. However, it remains open to further inquiry how much these scores are constrained by the comparatively small capacity of our BabyLMs, and whether larger models would enhance performance further.

B Data creation and tokenization analysis

Data creation Word frequency influences lexical decision performance greatly (McClelland and Rumelhart, 1981; Allen et al., 2005). To incorporate this effect into our study, we create two distinct data sets from words included in `wuggy`: (i) high-frequency stimuli with a frequency score over 7.0 (occurrences per 1M words in BNC, COCA and other English corpora, as reported in CELEX, Baayen et al., 1995) and (ii) low-frequency stimuli with a frequency score over 0.0 but below 0.7 (so at least one order of magnitude lower). We opted to rely on the CELEX frequency scores because we compare models trained on different corpora – our models are trained on the 10M BabyLM corpus, the models by Goriely et al. (2024) are trained on the 100M BabyLM corpus and the Pythia models are trained on The Pile (Gao et al., 2020). As such, frequency scores from these corpora would taint analyses of other models and hinder comparability. For the contextualised stimuli, we sample sentences from the OpenSubtitles (Lison and Tiedemann, 2016) portion of the BabyLM 2024 corpus (Choshen et al., 2024). Both Wuggy and BabyLM data are licensed under the MIT license[5], therefore we release our own stimuli artifacts on HuggingFace[6] under the same license. The data contain no information that names or uniquely identifies individual people or offensive content, and are commonly used in computational linguistics.

Analysis of tokenization To further assess the influence that these frequency scores have on the resulting tokenization for our own models, we offer a brief analysis: Figure 4 shows a pairplot between three numerical variables – (i) the number of tokens that our original words are split into, (ii) the number of tokens that the derived non-word are split into and (iii) the corresponding frequency score from CELEX. While the three plots on the diagonal axis show a layered kernel density estimate (KDE) for

[4] https://huggingface.co/collections/bbunzeck/word-learning-in-small-lms-67bdc218688856162b3be08f

[5] As per license information found at https://github.com/WuggyCode/wuggy and https://github.com/babylm/evaluation-pipeline-2024.

[6] https://huggingface.co/datasets/bbunzeck/lexical-decision

	small-char	medium-char	large-char	small-bpe	medium-bpe	large-bpe
Embedding size	128	256	512	128	256	512
Hidden size	128	256	512	128	256	512
Layers	4	8	12	4	8	12
Attention heads	4	8	12	4	8	12
Context size	128	128	128	128	128	128
Vocab. size	102	102	102	8,002	8,002	8,002
Parameters	486,016	3,726,592	21,940,736	2,508,416	7,771,392	30,030,336

Table 2: Model hyperparameters for our self-trained Llama models

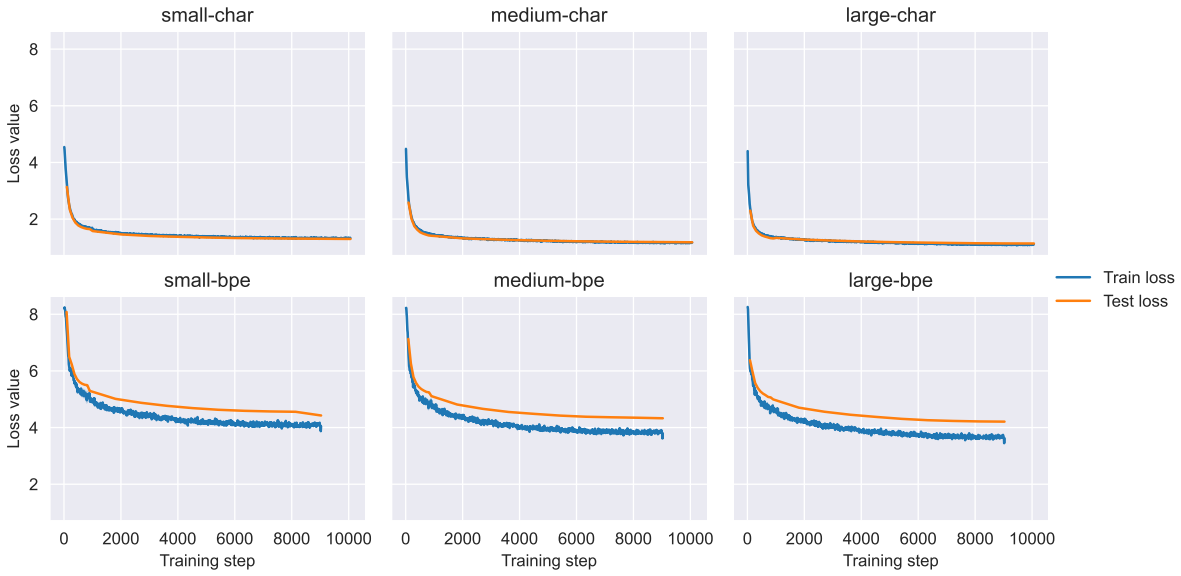


Figure 3: Loss curves for our self-trained Llama models

each individual variable, the other plots are scatterplots which visualize the relationship between the variables. The data points are colored for their tokenization scheme.

In the upper left and lower right plot we can see that both real and non-words are split similarly in the two distinct tokenization schemes. Words split by the BPE tokenizer tend to have fewer tokens, mostly between one and six. For the character-based tokenizer, a normal distribution is visible, with its peak at six tokens.

The upper right and lower left scatter plots show the relationship between tokenization for real and non-words. The character-level tokenization exhibits perfect alignment between both kinds of stimuli, they are always split into the exact same number of tokens. Subword tokenization is slightly skewed towards the non-word tokens. This means that non-words are more often split into more tokens than real words, although the reverse case is not completely infrequent.

C Full learning curves for BLiMP and word learning

Figure 5 shows the full learning curves of all phenomena included in BLiMP (individual syntactic paradigms belonging to one phenomenon are displayed in the same sub-figure) as well as our own lexical benchmarks (all displayed in individual sub-figures), for our six self-trained models and the six Pythia models that we compare them to. We fit a fifth-order polynomial curve to the individual data points and display it on a logarithmic scale.

It should be noted that we plot the number of the checkpoint on the x-scale. However, the individual amounts of actual textual data seen between these checkpoints differs vastly between our self-trained models (10M lexical tokens) and the Pythia models (825GB of textual data; as the dataset has since been taken down, no lexical token counts are possible anymore).

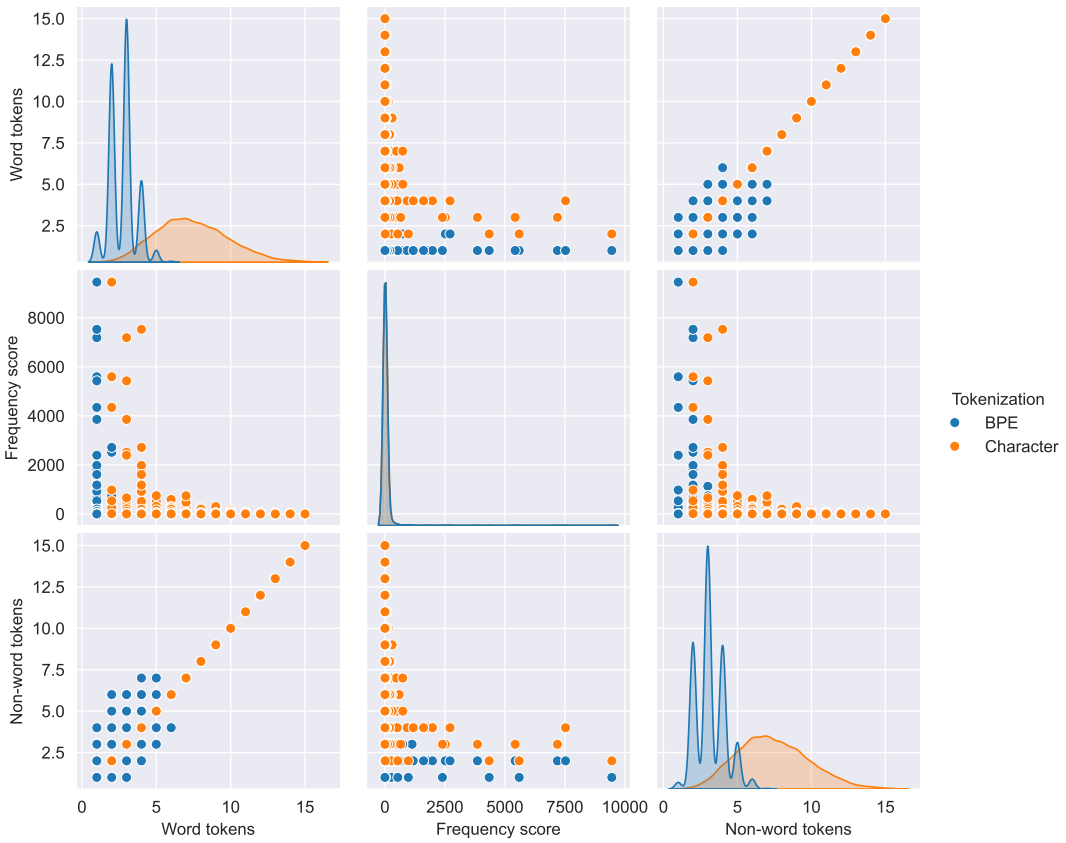


Figure 4: Pairplot displaying (i) number of tokens of words, (ii) frequency scores from CELEX (Baayen et al., 1995) and (iii) number of tokens of non-words (for BPE and character tokenization)

D Correlations between word learning and syntactic learning

As an additional measure of commonalities between word learning and syntactic learning, we calculate Spearman-rank correlation scores between ordered accuracy scores for our lexical tasks and BLiMP paradigms. Table 3 shows the underlying numerical values for the correlation heatmap provided in Figure 6 (please note that the heatmap is rotated in comparison to the table). All scores are statistically significant ($p < 0.05$). Due to the similar learning curves found in Figure 2, we average accuracy scores over all lexical phenomena (lexical decision and both surprisal settings), and then calculate correlations between them (both high and low frequency) and the coarse-grained BLiMP phenomena. For BPE models, lexical performance is highly correlated with more than half of the BLiMP phenomena. The character models show much weaker correlation with syntactic learning. This further confirms our findings about the strong entanglement of lexical and syntactic learning in subword models and their weaker ties in character models.

E Final BLiMP scores for all models

We reproduce the final syntactic evaluation scores for all models that we incorporated in our lexical analyses in Table 4. Generally, scores improve with larger models and with more training data. Most strikingly, subword models are consistently superior to comparable character models trained on the same amount of data. These differences, however, are most pronounced for the small models trained on very little data, like our Llama models trained on 10M tokens (7% for smallest models, 2% for largest models). For the comparable GPT-2 models trained on 100M tokens, the gap becomes much smaller (0.4%).

F Development of word/non-word differences

In Figure 7, we plot the average difference between word and non-word negative log-probability values across training, for both high-frequency and low-frequency data. Positive scores indicate preference

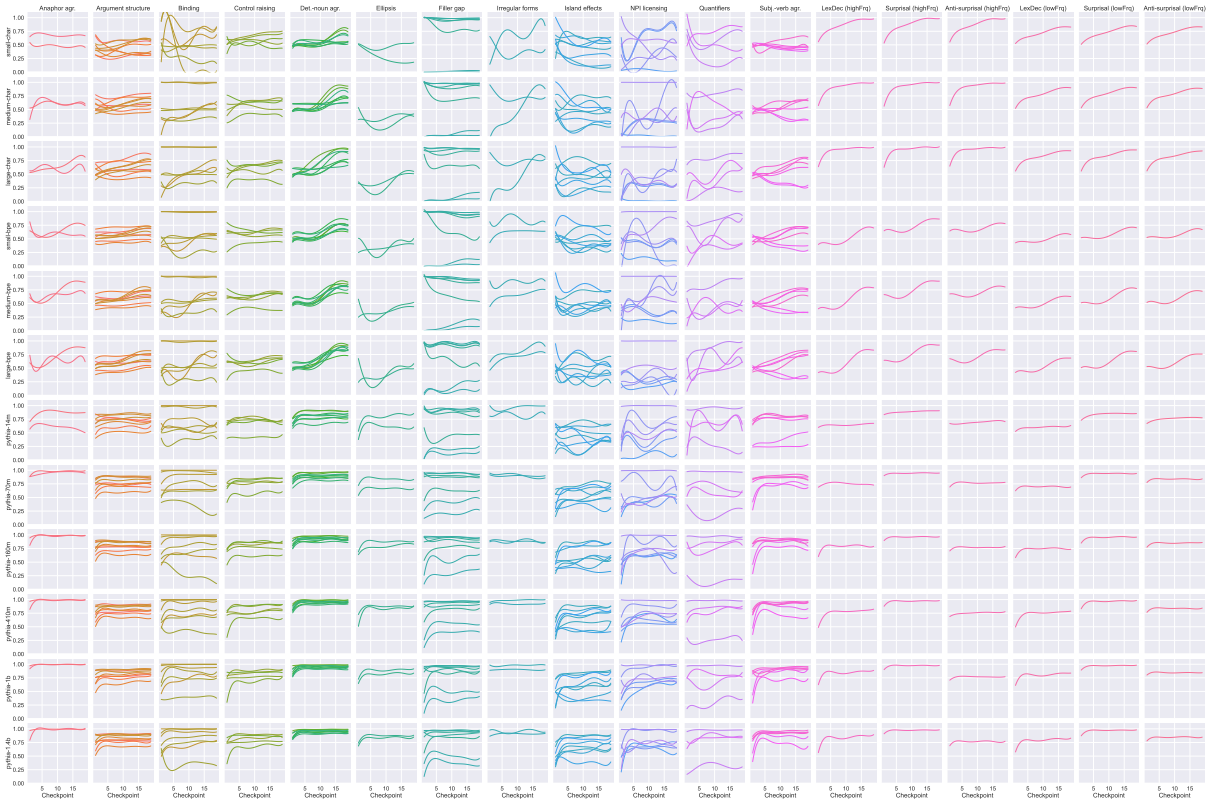


Figure 5: Learning curves for all paradigms in BLiMP and high/low frequency lexical decision data, separated for models (rows) and phenomenon sets (columns)

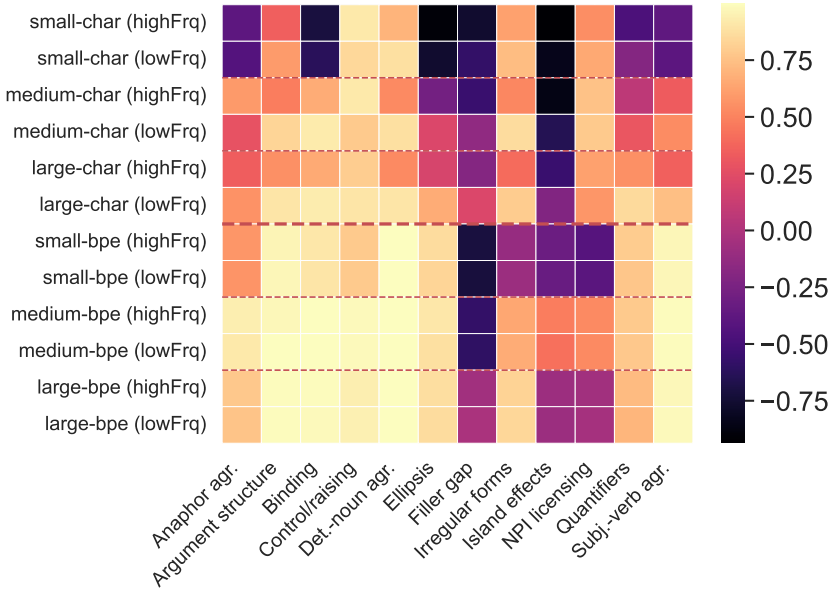


Figure 6: Correlation heatmap

for real words. For the character models, the differences are generally less pronounced and get most extreme at the end of pre-training (where accuracy scores do not change anymore), especially for the lexical decision data, which is already consistent at very early training stages. For the BPE models, we see that at the beginning they actually prefer non-words in the lexical decision task. Only after the first 10% of training they *begin* to discern words and non-words. While overall tendencies remain the same for both frequency conditions, the absolute differences are generally lower and the

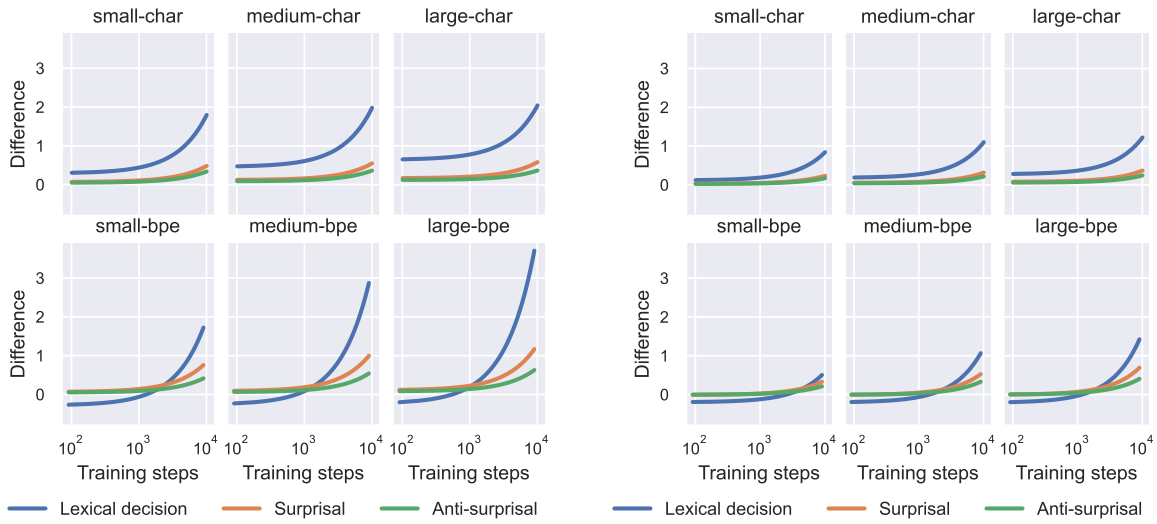
BLiMP phenomenon	small-char		medium-char		large-char		small-bpe		medium-bpe		large-bpe	
	highFrq	lowFrq	highFrq	lowFrq	highFrq	lowFrq	highFrq	lowFrq	highFrq	lowFrq	highFrq	lowFrq
Anaphor agr.	-0.393	-0.435	0.580	0.277	0.332	0.555	0.569	0.559	0.930	0.910	0.772	0.759
Argument structure	0.339	0.589	0.467	0.825	0.545	0.895	0.949	0.959	0.970	0.987	0.979	0.986
Binding	-0.718	-0.629	0.660	0.918	0.653	0.922	0.901	0.889	0.992	0.993	0.979	0.977
Control raising	0.904	0.837	0.909	0.776	0.791	0.892	0.777	0.780	0.974	0.974	0.930	0.936
Det.-noun agr.	0.686	0.870	0.524	0.869	0.521	0.890	0.989	0.989	0.990	0.993	0.994	0.989
Ellipsis	-0.912	-0.766	-0.285	0.209	0.180	0.662	0.857	0.822	0.897	0.868	0.865	0.856
Filler gap	-0.765	-0.586	-0.554	-0.146	-0.200	0.209	-0.715	-0.722	-0.589	-0.602	-0.063	-0.031
Irregular forms	0.612	0.724	0.507	0.856	0.397	0.787	-0.116	-0.098	0.636	0.662	0.816	0.832
Island effects	-0.937	-0.840	-0.862	-0.657	-0.556	-0.214	-0.321	-0.334	0.473	0.418	-0.088	-0.094
NPI licensing	0.535	0.641	0.748	0.782	0.616	0.568	-0.425	-0.407	0.520	0.526	-0.069	-0.049
Quantifiers	-0.476	-0.202	0.059	0.299	0.547	0.848	0.789	0.767	0.779	0.770	0.716	0.695
Subj.-verb agr.	-0.386	-0.400	0.329	0.529	0.351	0.730	0.963	0.961	0.982	0.981	0.967	0.974

Table 3: Spearman-rank correlation scores between ordered accuracy scores for our lexical tasks and BLiMP paradigms

Tok.	Model	Params	BLiMP score
Subword (BPE)	Pythia	14M	65.86%
		70M	73.30%
		160M	77.50%
		410M	81.63%
		1B	82.21%
		1.4B	81.92%
	GPT-2	85M	77.80%
	Llama	2.51M	59.80%
		7.77M	64.55%
		30.03M	64.56%
Character	GPT-2	85M	77.40%
	Llama	0.49M	52.69%
		3.73M	51.07%
		21.94M	62.14%

Table 4: BLiMP scores for all models

differences between the curves are less pronounced in the low-frequency setting.



(a) High-frequency data

(b) Low-frequency data

Figure 7: Average differences between surprisal values across pretraining

LLM as Entity Disambiguator for Biomedical Entity-Linking

Christophe Ye **Cassie S. Mitchell**

Georgia Institute of Technology

cye73@gatech.edu,
cassie.mitchell@bme.gatech.edu

Abstract

Entity linking involves normalizing a mention in medical text to a unique identifier in a knowledge base, such as UMLS or MeSH. Most entity linkers follow a two-stage process: first, a candidate generation step selects high-quality candidates, and then a named entity disambiguation phase determines the best candidate for final linking. This study demonstrates that leveraging a large language model (LLM) as an entity disambiguator significantly enhances entity linking models' accuracy and recall. Specifically, the LLM disambiguator achieves remarkable improvements when applied to alias-matching entity linking methods. Without any fine-tuning, our approach establishes a new state-of-the-art (SOTA), surpassing previous methods on multiple prevalent biomedical datasets by up to 16 points in accuracy. We released our code on GitHub at https://github.com/ChristopheYe/llm_disamb.

1 Introduction

The biomedical domain is an information-rich and highly specialized field, characterized by vast amounts of domain-specific knowledge and intricate terminologies. Unlike named entity recognition (NER), which focuses on identifying entity mentions in text, biomedical entity linking (EL) goes a step further by mapping these mentions to unique identifiers in a structured knowledge base (KB). This distinction is critical, as biomedical texts often contain extensive synonyms, polysemy, and abbreviations, where the same concept may be expressed in multiple ways, or a single term may refer to different entities depending on context. Effective EL is essential for resolving these ambiguities, enabling more precise information retrieval (Lee et al., 2016), improving knowledge discovery (Wang et al., 2018), and facilitating downstream tasks such as automated data annotation and curation. By strengthening EL systems, researchers can unlock more efficient data integration, enhance text mining applications, and accelerate advancements in biomedical research and healthcare.

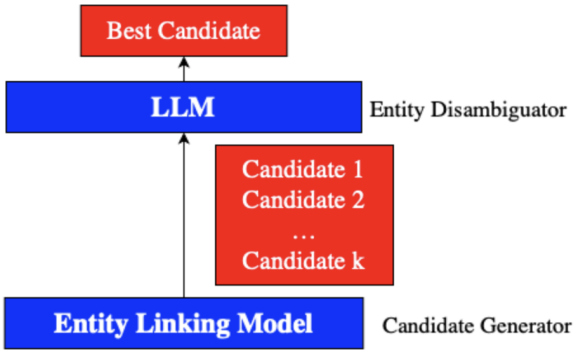


Figure 1: LLM as Entity Disambiguator.

Various methods have been applied to biomedical EL. Alias-matching approaches align mentions with entities based on lexicographical properties. This includes models like SciSpacy (Neumann et al., 2019), MetaMap (Aronson and Lang, 2010) or SapBERT (Liu et al., 2021). Contextualized EL learns vector representations (Vaswani et al., 2023) of entities by leveraging the contextual information from the mention and try to match it to an entity description. This includes models like KRISSBERT (Zhang et al., 2022), ClusterEL (Angell et al., 2021) or ArboEL (Agarwal et al., 2022). These representations are typically employed to generate a list of high-quality candidates for a mention, which are then refined using a re-ranker (Wu et al., 2020).

Large language models (LLMs) have recently shown promise in various biomedical Natural Language Processing (NLP) (Tian et al., 2023) applications. However, for information extraction tasks like entity recognition, their performance still lags significantly behind the previously cited models (Jahan et al., 2024).

Prior work has leveraged LLMs as external tools

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 301–312
July 27 - August 1, 2025

to enhance data quality for improved entity normalization. For instance, Borchert et al. (2024) simplifies entity mentions by rephrasing long or complex terms into more general or concise forms before candidate generation, while Chen employs LLMs to generate high-quality augmented data, improving the performance of traditional biomedical entity linking models. Garda and Leser (2024) integrates LLM-powered homonym disambiguation, which resolves ambiguous entity names by appending clarifying descriptors, and candidate sharing, which enhances training by leveraging contextually related mentions within the same document. However, no prior work has fully integrated LLMs as a core component of the entity linking pipeline itself.

This is primarily due to a lack of domain-specific knowledge and tendencies toward hallucination. However, with sufficient information, LLMs could potentially leverage their general understanding of text to perform the normalization. In this work, we explore the use of LLMs for biomedical entity disambiguation task.

This paper contributes the following :

- LLM is introduced as a **entity disambiguator** for named entity disambiguation, building on top of the existing methods for candidate generation.
- LLM entity disambiguation achieved consistent improvements for alias-matching EL models. It outperformed previous SOTA for several datasets with a remarkable gain of 16 points in GNormPlus.
- The method can be seamlessly integrated into existing EL models without requiring additional training.

2 Methodolody

2.1 Entity Linking model for candidate generation

The first step of our approach involves generating candidates using EL models. While multiple models are available, we will concentrate on the top-performing model in each category: SapBERT for alias-matching EL and ArboEL for contextualized EL.

SapBERT (Liu et al., 2021) selects top candidates by computing cosine similarity between the test mention and all aliases in the database, using enhanced embedding representations. ArboEL (Agarwal et al., 2022) first trains a bi-encoder to improve embedding representations using mention–mention coreference signals. This is followed by a cross-encoder training that scores each candidate by concatenating the test mention with the candidate's context. The top-k candidates are then selected based on these scores.

2.2 Disambiguation step with LLMs

2.2.1 The mention

To fully leverage the capabilities of the LLM, it's crucial to provide the mention along with its surrounding context, ensuring the model has access to the maximum amount of relevant information. We formatted the mention for the LLM as follows:

$$c_{left} \text{ [ENTITY START] } m \text{ [ENTITY END] } c_{right}$$

where c_{left} and c_{right} represent the left and right contexts of the mention m.

2.2.2 The entity

Similarly, to ensure maximum reliability, we provided the entity along with its relevant data to the LLM in the following dictionary format:

```
entity_data = {
    "cui": entity.cui,
    "name": entity.name,
    "types": entity.types,
    "aliases": entity.aliases,
    "definition": entity.definition,
}
```

- cui: Concept Unique Identifier
- name: Entity name
- types: Entity Types
- aliases: Entity aliases
- definition: Entity definition

2.2.3 In-context Learning

In-context learning (Dong et al., 2024) is an approach in natural language processing (NLP) where LLMs make predictions based on contexts enriched with a few examples. It has been shown that incorporating such examples in the prompt can enhance model performance. To optimize the selection of examples, we employed a RAG-like system (Lewis et al., 2021) that retrieves the most relevant examples from the training set to include in the prompt.

Specifically, the LLM predicts the response to the query x by conditioning on k training examples $\{(x_i, y_i)\}_{i=1}^{k}$ in the prompt :

$$\hat{y} = \text{LLM}(x|x_1, y_1, \cdots, x_k, y_k)$$

To find the most relevant examples for a given query, we first embed all mentions along with their surrounding context from the training set using a sentence embedding model (Gao et al., 2022). We then create an index using Faiss (Douze et al., 2024) for faster and more efficient retrieval. During inference, we embed the query (the mention along with its context) and retrieve the most similar mentions from the corpus for the LLM to process.

3 Results

The performance of LLMs as entity disambiguators was evaluated across multiple biomedical datasets. Prior work has typically relied on supervised learning using BERT-base models for entity linking tasks. In a comprehensive evaluation, Kartchner et al. (2023) identified SapBERT as the most effective alias-matching model, and ArboEL as the strongest contextualized model. These results were also later confirmed by Bathala et al. (2025) in their benchmarking package. Based on these evidences, our assessment is conducted by applying LLMs to these two leading approaches.

Performance was evaluated using five different models : Meta-Llama-3.1-8B-Instruct (Touvron et al., 2023), Mistral-Nemo-Instruct-2407 (Jiang et al., 2024), Qwen2.5-7B-Instruct (Bai et al., 2023), GPT-4o-mini and GPT-4o (team, 2024).

All LLMs were prompted using a greedy-decoding strategy; temperature was set to 0.

A comprehensive description of the experimental setup is provided in Appendix A.1 to ensure reproducibility of the experiments. This includes the exact details of the prompt used to obtain the correct CUI and the top-k candidates (Appendix A.1.1), along with the analysis of the impact of the number of candidates (Appendix A.1.3) and number of examples (Appendix A.1.4) to include in the prompt.

3.1 Datasets

The experiments were carried out on five prevalent biomedical datasets coming from BigBio (Fries et al., 2022) : NCBI-Disease (Dogan et al., 2014), GNormPlus (Wei et al., 2015), NLM Chem (Dogan et al., 2021a), NLM Gene (Dogan et al., 2021b), and Medmentions-ST21PV (Mohan and Li, 2019).

Details on the datasets are provided in Table 1.

The corresponding ontologies are MEDIC (Davis et al., 2019), Entrez (Maglott et al., 2005), MeSH (Lipscomb, 2000), and UMLS (Bodenreider, 2004).

Dataset	Train Mentions	Test Mentions	Ontology
NCBI-Disease	6,881	960	MEDIC
GNormPlus	6,252	3,223	Entrez
NLM-Chem	37,999	11,660	MeSH
NLM-Gene	15,553	2,729	Entrez
MM-ST21PV	203,282	40,143	UMLS

Table 1: Datasets used for evaluation.

3.2 With Alias-Matching EL

The impact of LLMs as entity disambiguators is illustrated for alias-matching EL methods using SapBERT. All results are detailed in Table 2.

When the base model performs well, with accuracy and recall@20 being close, the LLM's impact is minimal. For instance, on the NCBI-Disease dataset, accuracy improves slightly, while recall@5 remains unchanged.

However, when the initial model struggles, integrating an LLM can significantly boost performance. For instance, SapBERT fails to effectively differentiate aliases in gene-centric datasets like GNormPlus and NLM-Gene, as indicated by the large gap between its accuracy and recall@20. Using GPT-4o as a disambiguator improved SapBERT's accuracy on GNormPlus from 19.1% to 74.8% and recall@5 from 56.6% to 83.8%, surpassing the previous SOTA by 16% in accuracy and 19% in recall@5.

SapBERT struggles to distinguish between multiple gene entities sharing the same alias. However, since candidates with the same alias are ranked closely, the correct entity often appears shortly after in the list. By leveraging contextual information, the LLM can distinguish the correct entity more efficiently addressing this limitation.

GPT-4o is the top-performing disambiguator for alias matching EL models, outperforming smaller 7B-12B models. However, it is noteworthy that these "small" open-source LLMs still provide significant improvements and even surpass the previous state-of-the-art (SOTA) in many scenarios.

The reported SOTA numbers are based on the results we obtained from re-running the models ourselves using the BioEL package with default parameters (Bathala et al., 2025), not the ones originally reported by (Kartchner et al., 2023).

	Base: SapBERT			Base model + Llama3		Base model + Mistral		Base model + Qwen2.5		Base model + GPT-4o-mini		Base model + GPT-4o		SOTA	
	@1	@5	@20	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5
NCBI-Disease	0.752	0.899	0.924	0.781 ↑	0.835 ↓	0.782 ↑	0.852 ↓	**0.791** ↑	**0.899**	0.783↑	0.887 ↓	0.790 ↑	0.869 ↓	0.771	0.820
GNormPlus	0.191	0.566	0.862	0.444 ↑	0.780 ↑	0.482 ↑	0.770 ↑	0.365 ↑	0.772 ↑	0.386 ↑	0.761 ↑	**0.748** ↑	**0.838** ↑	0.585	0.647
NLM-Chem	0.754	0.876	0.889	0.842 ↑	0.876	0.833 ↑	0.879 ↑	0.849 ↑	0.876	0.849 ↑	0.879 ↑	**0.859** ↑	**0.879** ↑	0.790	0.856
NLM-Gene	0.072	0.344	0.824	0.314 ↑	0.659 ↑	0.324 ↑	0.571 ↑	0.251 ↑	0.599 ↑	0.251 ↑	0.556 ↑	0.497 ↑	0.668 ↑	**0.559**	**0.751**
MM-ST21PV	0.594	0.771	0.794	0.667 ↑	0.776 ↑	0.664 ↑	0.787 ↑	0.675 ↑	0.785 ↑	N/A	N/A	N/A	N/A	**0.685**	**0.798**

Table 2: Comparison Accuracy (Recall@1) and Recall@5 between initial base model SapBERT and after applying different LLM disambiguators. N/A : Not Available due to prohibitive cost considerations. SOTA model is ArboEL.
x : Best result / x : Beat SOTA / ↑ : Improvement over base / ↓ : Degradation from base

	Base: ArboEL			Base model + Llama3		Base model + Mistral		Base model + Qwen2.5		Base model + GPT-4o-mini		Base model + GPT-4o	
	@1	@5	@20	@1	@5	@1	@5	@1	@5	@1	@5	@1	@5
NCBI-Disease	0.771	0.820	0.838	0.760 ↓	0.816 ↓	0.749 ↓	0.821 ↑	0.767 ↓	0.822 ↑	0.764 ↓	0.833 ↑	0.758 ↓	**0.837** ↑
GNormPlus	0.585	0.647	0.659	0.580 ↓	0.649 ↑	0.585	0.646 ↓	0.600 ↑	0.647	0.600 ↑	0.652 ↑	**0.618** ↑	**0.654** ↑
NLM-Chem	0.790	**0.856**	0.863	0.815 ↑	0.849 ↓	0.806 ↑	0.847 ↓	0.821 ↑	0.849 ↓	**0.830** ↑	0.853 ↓	0.815 ↑	0.852 ↓
NLM-Gene	**0.559**	**0.751**	0.778	0.549 ↓	0.700 ↓	0.549 ↓	0.714 ↓	0.550 ↓	0.726 ↓	0.538 ↓	0.736 ↓	0.537 ↓	0.726 ↓
MM-ST21PV	0.685	**0.798**	0.811	0.649 ↓	0.735 ↓	0.641 ↓	0.767 ↓	**0.686** ↑	0.790 ↓	N/A	N/A	N/A	N/A

Table 3: Comparison Accuracy (Recall@1) and Recall@5 between initial base model ArboEL and after applying different LLM disambiguators. N/A : Not Available due to prohibitive cost considerations. ArboEL is SOTA model.
x : Best result / x : Beat SOTA / ↑ : Improvement over base / ↓ : Degradation from base

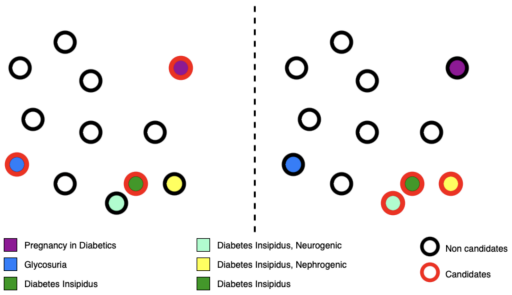


Figure 2: Embedding Space of Candidates from Alias Matching (left) and Contextualized EL model (right)

3.3 With Contextualized EL

The impact of LLMs as entity disambiguators was demonstrated for contextualized EL methods using ArboEL. The results are detailed in Table 3.

When applied to contextualized EL, the results show minimal improvement for the best model and even degrade for others. This happens because the base model already handles mentions effectively when the correct CUIs are included in the candidate list, as it is also considering the contextual information of the mention.

3.4 Performance disparity

The performance gap between contextualized and alias-matching EL models stems from their differing approaches. Alias-matching methods often achieve higher recall at large recall levels by capturing a broad range of potential matches based on surface similarities (e.g., aliases or synonyms), without filtering out even when the candidates are totally different.

Context-aware models, while generally more precise, tend to offer less improvement in recall@k at larger k values. These models are designed to sharply distinguish between correct and incorrect entities, creating a large gap between top-ranked candidates and those deemed contextually different. While this helps in pinpointing the correct entity, it often discards candidates that deviate from the expected answer. As a result, if the correct answer is not initially ranked high due to contextual ambiguity, it may be pushed far down the ranking for appearing too dissimilar.

Consequently, context-aware models may miss the correct candidate in large pools, where alias-matching methods would still retain it, even if ranked lower.

Figure 2 provides an example of such scenario :

while alias-matching EL produced a diverse set of top-ranked candidates, (e.g., "Pregnancy in Diabetics," "Glycosuria," "Diabetes Insipidus") contextualized EL generated candidates that were more closely related and grouped together in the rankings. (e.g., "Diabetes Insipidus," "Diabetes Insipidus, Neurogenic," "Diabetes Insipidus, Nephrogenic")

Further results on error analysis, performance disparity in low-data slices and data leakage risks are available in Appendix A.2

4 Conclusion and Future Work

Entity linking is crucial in knowledge-driven NLP, particularly in scientific and biomedical domains, where accurately mapping textual mentions to specific concepts is essential for extracting meaningful insights and advancing research. This study introduced a novel approach that utilized LLMs as entity disambiguators for biomedical EL. The proposed methodology demonstrated a significant performance improvement when integrated with an alias-matching-based EL model, as evidenced by experiments conducted on five standard biomedical EL datasets. Notably, this approach requires no fine-tuning, making it highly adaptable as LLMs continue to evolve in robustness and scalability. A promising direction for future work is the development of a retrieval mechanism to identify additional high-quality candidates when the correct entity is absent from the initial candidate set due to omissions by the underlying EL model.

5 Limitations

This method has demonstrated substantial effectiveness in enhancing the performance of alias-matching EL models. However, its application to contextualized EL models can occasionally result in performance degradation, leading to inconsistencies in outcomes. Furthermore, leveraging LLMs for entity disambiguation introduces a significant trade-off in inference time, which poses potential scalability challenges for datasets containing millions of mentions. The approach is also highly dependent on the candidate generation step; if the correct Concept Unique Identifier (CUI) is absent from the candidate set, there is no mechanism to retrieve it. Additionally, the use of proprietary LLMs can be prohibitively expensive. Finally, LLMs remain susceptible to errors, a critical concern in biomedical applications where precision is essential and any degree of randomness is unacceptable.

Acknowledgments

Funding support provided by National Science Foundation CAREER grant 1944247, National Institute of Health grant R35GM152245, and Chan Zuckerberg Initiative grant 253558 to C.S.M.

References

Dhruv Agarwal, Rico Angell, Nicholas Monath, and Andrew McCallum. 2022. Entity linking via explicit mention-mention coreference modeling. pages 4644–4658.

Rico Angell, Nicholas Monath, Sunil Mohan, Nishant Yadav, and Andrew McCallum. 2021. Clustering-based inference for biomedical entity linking. *Preprint*, arXiv:2010.11253.

Alan R Aronson and François-Michel Lang. 2010. An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Prasanth Bathala, Christophe Ye, Batuhan Nursal, Shubham Lohiya, David Kartchner, and Cassie S. Mitchell. 2025. BioEL: A comprehensive python package for biomedical entity linking. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1709–1721, Albuquerque, New Mexico. Association for Computational Linguistics.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Florian Borchert, Ignacio Llorca, and Matthieu-P Schapranow. 2024. Improving biomedical entity linking for complex entity mentions with llm-based text simplification. *Database*, 2024:baae067.

Haihua Chen.

Allan Peter Davis, Cynthia J Grondin, Robin J Johnson, Daniela Sciaky, Roy McMorran, Jolene Wiegers, Thomas C Wiegers, and Carolyn J Mattingly. 2019. The comparative toxicogenomics database: update 2019. *Nucleic acids research*, 47(D1):D948–D954.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, and more Deepseek team. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Rezarta Dogan, Robert Leaman, Sun Kim, Dongseop Kwon, Chih-Hsuan Wei, Donald Comeau, Yifan Peng, David Cissel, Cathleen Coss, Carol Fisher, Rob Guzman, Preeti Kochar, Stella Koppel, Dorothy Trinh, Keiko Sekiya, Janice Ward, Deborah Whitman, Susan Schmidt, and Zhiyong lu. 2021a. Nlm-chem, a new resource for chemical entity recognition in pubmed full text literature. *Scientific Data*, 8.

Rezarta Dogan, Chih-Hsuan Wei, David Cissel, Nicholas Miliaras, Olga Printseva, Oleg Rodionov, Keiko Sekiya, Janice Ward, and Zhiyong lu. 2021b. Nlm-gene, a richly annotated gold standard dataset for gene entities that addresses ambiguity and multi-species gene recognition. *Journal of Biomedical Informatics*, 118:103779.

Rezarta Islamaj Dogan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. *Preprint*, arXiv:2301.00234.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *Preprint*, arXiv:2401.08281.

Jason A. Fries, Troy Mayfield, Kenneth Brimacombe, Michael M. Bronstein, David Silver, David Wehner, et al. 2022. Bigbio: A large-scale biomedical corpus. *Journal of Biomedical Informatics*, 135:104037.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings. *Preprint*, arXiv:2104.08821.

Samuele Garda and Ulf Leser. 2024. Belhd: improving biomedical entity linking with homonym disambiguation. *Bioinformatics*, 40(8):btae474.

Israt Jahan, Md Tahmid Rahman Laskar, Chun Peng, and Jimmy Xiangji Huang. 2024. A comprehensive evaluation of large language models on benchmark biomedical text processing tasks. *Computers in Biology and Medicine*, 171:108189.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

David Kartchner, Jennifer Deng, Shubham Lohiya, Tejasri Kopparthi, Prasanth Bathala, Daniel Domingo-Fernández, and Cassie Mitchell. 2023. A comprehensive evaluation of biomedical entity linking models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14462–14478, Singapore. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180.

Sunwon Lee, Donghyeon Kim, Kyubum Lee, Jaehoon Choi, Seongsoon Kim, Minji Jeon, Sangrak Lim, Donghee Choi, Sunkyu Kim, Aik Choon Tan, and Jaewoo Kang. 2016. Best: Next-generation biomedical entity search tool for knowledge discovery from biomedical literature. *PLOS ONE*, 11.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Carolyn E Lipscomb. 2000. Medical subject headings (mesh). *Bulletin of the Medical Library Association*, 88(3):265.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238, Online. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Donna Maglott, Jim Ostell, Kim D Pruitt, and Tatiana Tatusova. 2005. Entrez gene: gene-centered information at ncbi. *Nucleic acids research*, 33(suppl_1):D54–D58.

Sunil Mohan and Donghui Li. 2019. Medmentions: A large biomedical corpus annotated with umls concepts. *Preprint*, arXiv:1902.09476.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

OpenAI team. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

S Tian, Q Jin, L Yeganova, et al. 2023. Opportunities and challenges for chatgpt and large language models in biomedicine and health. arxiv. *arXiv preprint arXiv:2306.10070*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Yanshan Wang, Sijia Liu, Naveed Afzal, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Paul Kingsbury, and Hongfang Liu. 2018. A comparison of word embeddings for the biomedical natural language processing. *Preprint*, arXiv:1802.00400.

Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong lu. 2015. Gnormplus: An integrative approach for tagging genes, gene families, and protein domains. *BioMed research international*, 2015:918710.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval.

Hongyi Yuan, Zheng Yuan, Ruyi Gan, Jiaxing Zhang, Yutao Xie, and Sheng Yu. 2022a. BioBART: Pretraining and evaluation of a biomedical generative language model. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 97–109, Dublin, Ireland. Association for Computational Linguistics.

Hongyi Yuan, Zheng Yuan, and Sheng Yu. 2022b. Generative biomedical entity linking via knowledge base-guided pre-training and synonyms-aware fine-tuning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4038–4048, Seattle, United States. Association for Computational Linguistics.

Sheng Zhang, Hao Cheng, Shikhar Vashishth, Cliff Wong, Jinfeng Xiao, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2022. Knowledge-rich self-supervision for biomedical entity linking.

A Appendix

The appendix is organized into three parts:

- The detailed method implementation is presented in Appendix A.1.
- Additional results on the evaluation are presented in Appendix A.2.
- Discussion on LLM for candidate generation A.3

A.1 Detailed Experimental Setup

This section outlines the experimental setup.

A.1.1 Prompts

Figures 3 and 4 illustrate the prompts used for the accuracy and recall tasks, respectively.

Elements enclosed in *{italics}* represent arguments passed to the prompt, which are detailed below:

- `number_candidates`: The number of candidates provided to the model.
- `mention`: The specific mention that needs to be correctly linked.
- `context`: The surrounding context in which the mention appears.
- `candidates`: The relevant data for all candidates provided to the model.
- `example_answers`: An example of valid answers.

A.1.2 Context window size

A 64-word context window was used, evenly split around each mention. This setup offered the best balance between performance and runtime.

A.1.3 Impact of the number of candidates in the prompt

We evaluated candidate set sizes ranging from 5 to 64 and found that using 10 to 20 candidates yielded the best performance. A set of only 5 candidates resulted in the exclusion of many correct options, whereas increasing the number to 50 introduced an excess of information, reducing precision as the LLM struggled to focus on relevant details, consistent with findings by (Liu et al., 2023).

Ultimately, the effectiveness of candidate selection depends on the ranking of correct entities. If a substantial portion of correct Concept Unique Identifiers (CUIs) are positioned beyond the top 20, increasing the number of candidates can enhance performance. However, in most cases—particularly for contextualized entity linking (EL) models—the proportion of correct candidates ranked below the top 20 is relatively low.

Figure 6 and 7 show the accuracy versus runtime for varying numbers of candidates in the prompt on GNormPlus and NLM-Gene dataset.

In the majority of configurations (dataset - CG Model - LLM), using 10 and 20 candidates yields the best performance.

A.1.4 Impact of the number of examples in the prompt

We also experimented with different numbers of examples in the prompt to evaluate their impact on performance.

Figure 8 and 9 presents the accuracy versus running time for varying numbers of examples k in the prompt on NCBI-Disease and GNormPlus dataset.

The optimal number of examples to include in the prompt varies depending on the configuration (dataset - CG Model - LLM). For instance, GNormPlus performance consistently improved as k increased, while NCBI-Disease performed best with k=3.

A.1.5 Use of separate prompts for accuracy and recall

The reported results in Table 2 and 3 for accuracy (recall@1) were all calculated independently, meaning the LLM was asked to return only the single most likely CUI for each query. Indeed, we noticed that the score for accuracy was higher when the LLM is tasked with retrieving a single candidate compared to ranking several candidates.

A.2 Additional results

This section presents additional results, including error analysis, data leakage risk assessment, a study of prompting strategies, statistical significance testing, and runtime evaluation.

A.2.1 Error Analysis

Table 4 presents the distribution of errors between Candidate Generation (CG) and Named Entity Disambiguation (NED) for each model. CG failure is defined as the absence of the correct CUI within the top-k generated candidates (in this case, k = 20). NED failure is defined as an instance where the

System Instruction
You are a professional data annotator and curator.
Your task is to identify the correct entity for a given mention based on the provided context and the descriptions of {*number_candidates*} candidate entities.

User
Here are a few examples:
{*topk_examples*}

This is the specific mention that needs to be linked to the correct entity:
{*mention*}

This is the context where the mention appears:
{*context*}

These are the candidate entities to choose from:
{*candidates*}

You MUST PROVIDE an ANSWER among the candidates
Use step-by-step reasoning but do not add provide any explanations to me. I only want the final answer.

Return the results in JSON format with just the CUI.
For instance {*example_answers*} are valid answers.

Figure 3: Prompt for the accuracy task, outputting only the best candidate CUI.

System Instruction
You are a professional data annotator and curator.
Your task is to rank the candidate entities from best to worst for a given mention based on the provided context and the descriptions of each candidate entities.

User
Here are a few examples:
{*topk_examples*}

This is the specific mention that needs to be linked to the correct entity:
{*mention*}

This is the context where the mention appears:
{*context*}

These are the candidate entities to choose from:
{*candidates*}

Rank the top {*recall_k*} candidate entities from best to worst.
Use step-by-step reasoning but do not add provide any explanations to me. I only want the final answer.

Return the results in JSON format as a list of CUIs ["CUI1", "CUI2", "CUI3", ...].
For instance {*example_answers*} are valid answers.

Figure 4: Prompt for the recall task, outputting only the best candidates CUI.

	NCBI-Disease		**GNormPlus**		**NLM-Chem**		**NLM-Gene**		**MM-ST21PV**	
	CG	ED	CG	ED	CG	NED	CG	ED	CG	ED
SapBERT	0.307	0.693	0.167	0.833	0.451	0.549	0.186	0.814	0.508	0.492
ArboEL	0.695	0.305	0.823	0.177	0.654	0.345	0.503	0.497	0.598	0.402

Table 4: Failure stage of the entity linking models.

top-ranked candidate is incorrect, when the correct CUI is indeed present in the list of plausible candidates. SapBERT is clearly better at identifying the correct alias than finding the correct candidate, as evidenced by its lower CG error and higher NED error. In this scenario, leveraging an LLM for the disambiguation step can significantly enhance performance, as demonstrated by the big improvement in gene-centric datasets.

A performance plot in low-data slices (training overlap, long-tail entities, no alias match, few aliases, zero-shot) before and after LLM disambiguation for NCBI-Disease, GNormPlus, NLM-Chem and NLM-Gene is shown in Figure 10.

ArboEL is observed to be more sensitive to training overlap and long-tail entities compared to SapBERT. ArboEL benefits greatly from prior exposure to similar examples in the training set across all datasets, while its performance deteriorates in zero-shot settings where either mentions or entities have not been encountered before (e.g., NLM-Chem, NCBI-Disease, NLM-Gene). When the mention does not match any alias of the correct CUI (no alias match), performance degradation is observed only in NCBI-Disease and NLM-Chem. Similarly, when the correct CUI has fewer than five aliases (<5 aliases), the effect is primarily noticeable in NCBI-Disease. Incorporating LLM-based disambiguation appears to mitigate these performance disparities across different data slices, leading to a more uniform performance across all datasets. This trend is consistent for both GPT-4o

System Instruction
You are a professional data annotator and curator.
Your task is to identify the correct entity for a given mention based on the provided context and the descriptions of {*number_candidates*} candidate entities.

User
Here are a few examples:
{*topk_examples*}

This is the specific mention that needs to be linked to the correct entity:
{*mention*}

This is the context where the mention appears:
{*context*}

These are the candidate entities to choose from:
{*candidates*}

You MUST PROVIDE an ANSWER among the candidates

Use step-by-step reasoning :
- Analyze the mention within the provided context to capture its intended meaning
- Evaluate each candidate entity based on its name, aliases, definition, and other attributes
- Pay special attention to semantic similarity whether the entity aligns with the mention's usage in context
- Discard entities that are inconsistent with the mention's meaning in context
- Among the remaining candidates, check the most precise match by considering finer details such as common usage and specific associations
- If multiple candidates remain viable, prioritize based on confidence alignment with context

Once you have reached a final decision, return the answer in the following JSON format.
For instance {*example_answers*} are valid answers.

Figure 5: Prompt for the accuracy task, with reasoning.

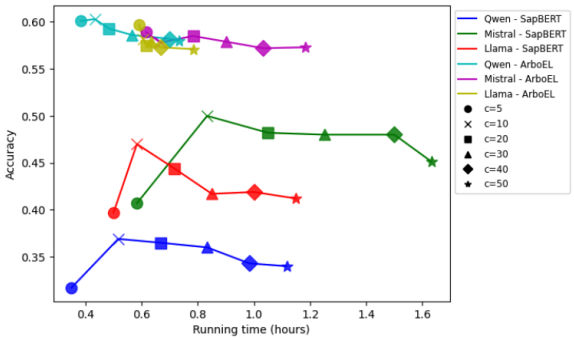


Figure 6: Accuracy vs running time for varying number of candidates in the prompt. Dataset : GNormPlus

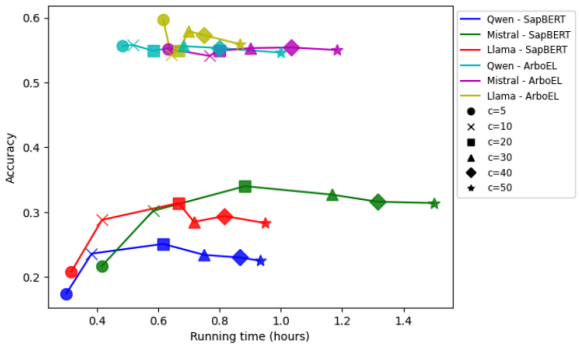


Figure 7: Accuracy vs running time for varying number of candidates in the prompt. Dataset : NLM-Gene

and Qwen, indicating that LLM-based disambiguation helps align slice-specific performance with the overall model performance.

A.2.2 Risk of data leakage

To provide more relevant context during inference, we selected the top-k most similar examples based on document-level similarity. While this approach improves contextual alignment, it also introduces the risk of data leakage (cases where the prompt includes the test mention). Such leakage can simplify the disambiguation task for the LLM.

To assess the extent and effect of this phenomenon, we analyzed how often it occurred and how it influenced model performance in Table 5. Our findings confirms that LLM indeed performs better when the prompt includes the test mention. In gene-centric datasets, the performance gain is more pronounced but these situations is very rare. In contrast, for datasets like NCBI-Disease, where such cases are more common, the performance difference is less significant.

We chose to retain these examples in the evaluation because such scenarios can plausibly arise in real-world applications.

A.2.3 Various prompts

The performance of three different prompts was evaluated using Qwen2.5 7B as base model :

- 1) A prompt designed to generate only the final candidate CUI without any explanation, enabling faster inference. (Figure 3)
- 2) A step-by-step reasoning prompt that first provides detailed reasoning before generating the final result, similar to chain-of-thought (Wei et al., 2023). (Figure 5)
- 3) A variation of the first prompt that utilizes the reasoning model DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI et al., 2025).

The results are presented in Figure 11. Among the evaluated prompting strategies, the simplest

	NCBI-Disease		GNormPlus		NLM-Chem		NLM-Gene	
	✓	✗	✓	✗	✓	✗	✓	✗
SapBERT + GPT-4o	0.842	0.767	0.862	0.742	0.880	0.853	0.582	0.492
SapBERT + Qwen2.5	0.838	0.773	0.661	0.360	0.871	0.836	0.363	0.245
Exact Test Mention present in few-shot	22.9%	77.1%	1.1%	98.9%	3.2%	96.8%	1.6%	98.4%

Table 5: Performance difference and frequency between cases with and without exact matches between prompt examples and the test mention.
✓: The test mention is in the few-shot examples / ✗ : The test mention is not in the few-shot examples

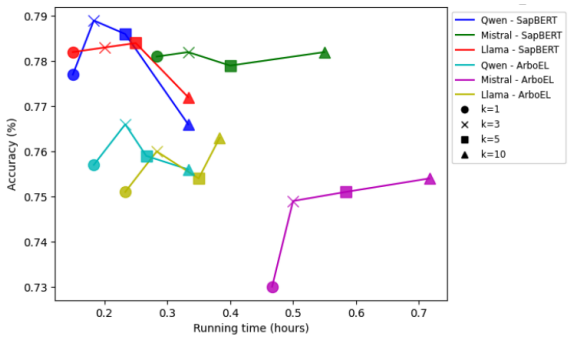


Figure 8: Accuracy vs Runtime for varying number of examples in the prompt. Dataset : NCBI-Disease

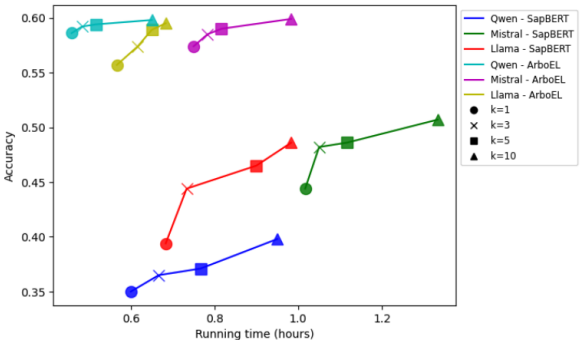


Figure 9: Accuracy vs Runtime for varying number of examples in the prompt. Dataset : GNormPlus

prompt—one that directly outputs the CUI without additional details—demonstrated the best performance across all datasets for this model. This approach not only proved to be significantly faster than the other two methods but also achieved higher accuracy. We hypothesize that this superiority stems from the nature of the task: the model can directly extract and infer the correct CUI from the given information without the need for multi-step reasoning. By avoiding unnecessary intermediate steps, it reduces the risk of error propagation and minimizes computational overhead.

A.2.4 Statistical significance tests

In Table 6 , we used McNemar's paired test to compute the p-values for the differences in accuracy between the base model and with LLMs across all evaluated datasets. The p-values determine whether the performance differences within each LLM are statistically significant across different datasets.

SapBERT combined with an LLM consistently show extremely low p-values across all datasets, indicating that the observed positive difference is statistically significant for all LLMs when applied on SapBERT.

These findings are consistent with the results presented in Section 3.

	SapBERT + Llama3	SapBERT + Mistral	SapBERT + Qwen2.5
NCBI-Disease	$9.14 \cdot 10^{-4}$	$4.51 \cdot 10^{-3}$	$5.38 \cdot 10^{-6}$
GNormPlus	$4.63 \cdot 10^{-175}$	$1.22 \cdot 10^{-217}$	$4.64 \cdot 10^{-132}$
NLM-Chem	$2.76 \cdot 10^{-212}$	$2.46 \cdot 10^{-158}$	$3.02 \cdot 10^{-188}$
NLM-Gene	$2.95 \cdot 10^{-152}$	$9.84 \cdot 10^{-163}$	$5.81 \cdot 10^{-102}$
MM-ST21PV	0.0	$8.30 \cdot 10^{-288}$	0.0

Table 6: P-value of Accuracy for all evaluated datasets and different LLMs using base model SapBERT.

A.2.5 Runtime

Figure 12 show the running time of various LLMs across different datasets, using SapBERT-generated candidates for Accuracy task.

Mistral, as the largest model, has the highest computational demand and requires the longest runtime. Running it on the MM-ST21PV dataset (31,827 evaluated mentions) took nearly a full day, underscoring the significant time requirements of this approach. As the dataset size increases, this can become prohibitively long, making this approach challenging for larger-scale applications.

However, this remains faster than ArboEL, which required 20 days to train on MM-ST21PV.

All experiments were run on a single Nvidia A40 GPU using vllm framework (Kwon et al., 2023).

A.3 LLM for candidate generation

Current LLMs face several limitations for direct candidate generation abilities. First, they lack explicit access to ontology databases, often leading

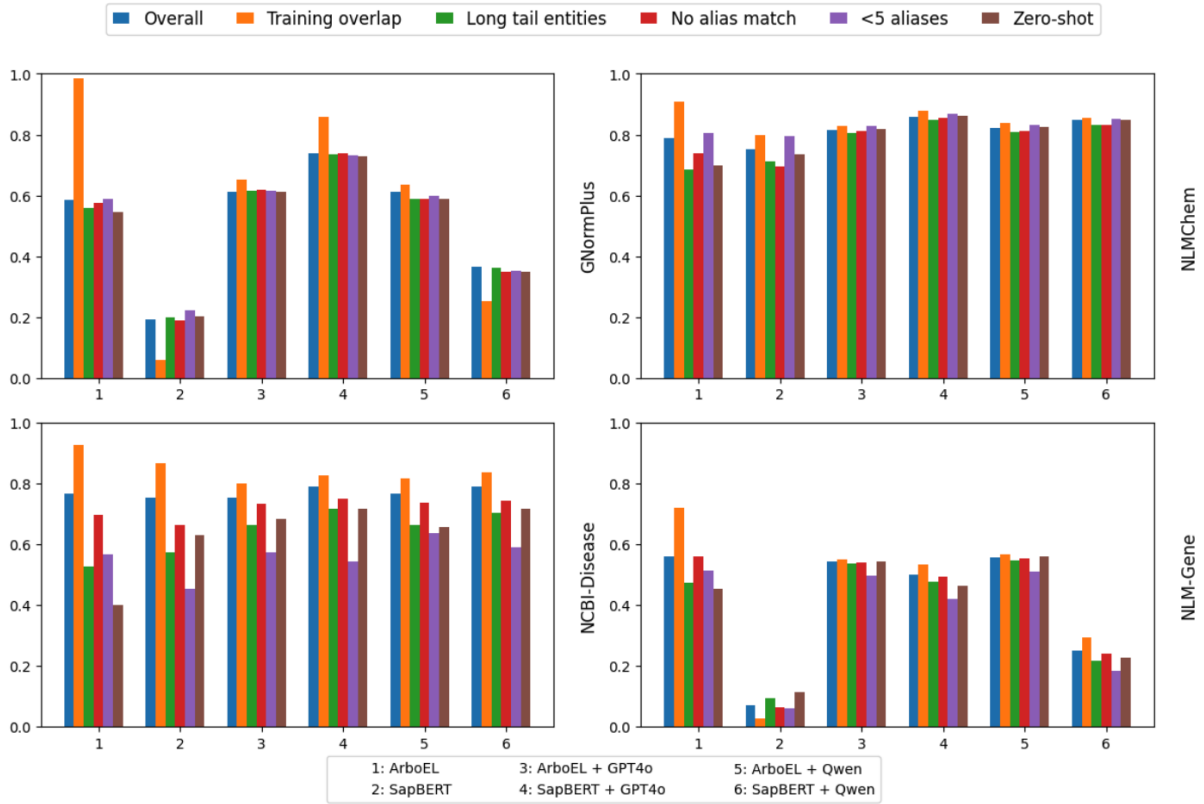


Figure 10: Performance across different data slices: training overlap, long-tail entities, no alias matches, few aliases, and zero-shot cases—both before and after LLM disambiguation.

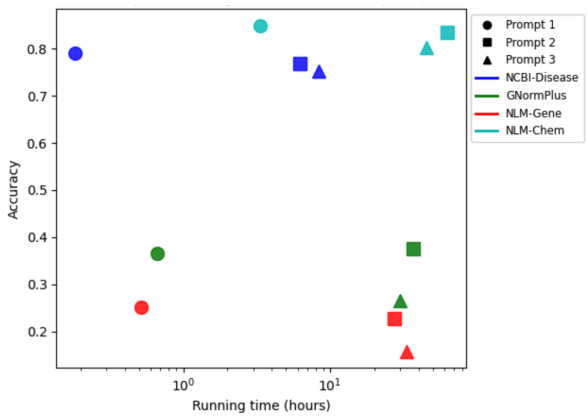


Figure 11: Accuracy vs. Runtime. Comparison of three prompting strategies across four datasets

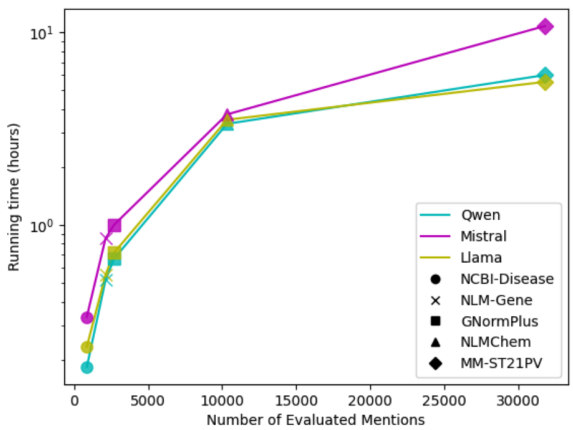


Figure 12: Runtime vs Number of Mentions for Accuracy task across Different LLMs. Base model = SapBERT

to hallucinations when generating candidates. Second, ontologies such as UMLS for MM-ST21PV contain over two million concepts, making it infeasible to include all candidates in the prompt due to context window constraints and computational overhead. These challenges make LLMs currently unreliable for direct large-scale candidate generation. Training models specifically for this task such as BioBART (Yuan et al., 2022a) or BioGenEL (Yuan et al., 2022b) could address the issue, but this lies beyond the scope of our work.

Towards Geo-Culturally Grounded LLM Generations

Piyawat Lertvittayakumjorn[⋆†], **David Kinney**[⋆†‡],
Vinodkumar Prabhakaran[†], **Donald Martin, Jr.**[†], **Sunipa Dev**[†]

[†]Google [‡]Washington University in St. Louis
{piyawat,vinodkpg,dxm,sunipadev}@google.com, kinney@wustl.edu

Abstract

Generative large language models (LLMs) have demonstrated gaps in diverse cultural awareness across the globe. We investigate the effect of retrieval augmented generation and search-grounding techniques on LLMs' ability to display familiarity with various national cultures. Specifically, we compare the performance of standard LLMs, LLMs augmented with retrievals from a bespoke knowledge base (i.e., KB grounding), and LLMs augmented with retrievals from a web search (i.e., search grounding) on multiple cultural awareness benchmarks. We find that search grounding significantly improves the LLM performance on multiple-choice benchmarks that test propositional knowledge (e.g., cultural norms, artifacts, and institutions), while KB grounding's effectiveness is limited by inadequate knowledge base coverage and a sub-optimal retriever. However, search grounding also increases the risk of stereotypical judgments by language models and fails to improve evaluators' judgments of cultural familiarity in a human evaluation with adequate statistical power. These results highlight the distinction between propositional cultural knowledge and open-ended cultural fluency when it comes to evaluating LLMs' cultural awareness.

1 Introduction

Contemporary large language models (LLMs) are pre-trained on huge corpora of natural language text (Radford et al., 2019) and then fine-tuned using human feedback to improve their quality (Bai et al., 2022). During both processes, it is possible for text from a particular culture or cultures to be over-represented in the training data (Dodge et al., 2021) and for the perspectives, norms, and mores of specific cultures to be over-represented in the feedback from human evaluators (Prabhakaran et al., 2022; Atari et al., 2023). Consequently, there is growing recognition of generative LLMs' shortcomings in representing and serving people from diverse geo-cultural backgrounds at the global scale (Adilazuarda et al., 2024; Pawar et al., 2024; Agarwal et al., 2025). Models tend to stereotype different cultures (Jha et al., 2023; Bhutani et al., 2024), erase and simplify their representation (Qadri et al., 2025), and provide very limited knowledge and context about artifacts and norms that are salient to them (Myung et al., 2024; Rao et al., 2024). Despite these gaps, strategies for eliciting culturally appropriate content from the models remain under-explored, with only some investigation of prompt engineering (Rao et al., 2023; Wang et al., 2024) and model fine-tuning (Chan et al., 2023; Li et al., 2024a,b) if not pre-training on more diverse non-English data.

Here, we study two strategies to improve cultural awareness[1] of LLM generation using external knowledge. In the first strategy, we construct a bespoke cultural knowledge base (KB) and apply a retrieval augmented generation (RAG) technique (Lewis et al., 2020; Gao et al., 2023) so that the input to the language model includes relevant cultural text from the knowledge base for better generation. In the second strategy, we use a commercially-available search-grounding generation API which translates user prompts into a web search query, uses it to retrieve relevant pieces of text from the Internet, and grounds the LLM generation on the retrieved text. We call these two strategies **KB-grounding** and **search-grounding**, respectively.

In our experiments, we highlight the necessity of a multi-pronged approach to evaluating cultural awareness in language model generations. Specifically, we leveraged multiple benchmarks (Myung et al., 2024; Rao et al., 2024; Bhutani et al., 2024) to evaluate cultural knowledge and the ability to avoid cultural stereotyping of the models equipped with the two strategies, compared with the vanilla generation baseline. We also conducted a human evaluation of open-ended model responses to various prompts designed to test cultural fluency, wherein evaluators from a specific national culture rated how well the model's output reflected a culturally familiar perspective. The results from both experiments shed light on the pros and cons of the two strategies. Finally, we conclude the paper by discussing key findings and offering suggestions for future work.

⋆ equal contribution

[1] We use the term "cultural awareness" to refer to the general ability of LLMs to work well across cultures. It is conceptually similar to other terms in the literature, e.g., "cultural appropriateness" and "cultural informedness." However, we are not attempting to define them precisely in this paper.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 313–330
July 27 - August 1, 2025

Source	Description	# Docs
CultureAtlas	Wikipedia text	239,376
Cube	Artefact names	198,896
CultureBank	Situation-based practices	22,990
SeeGULL	Stereotypes	6,871

Table 1: Sources of documents in our cultural KB.

2 Improving Cultural Awareness by Retrieving External Knowledge

Retrieval augmented generation (RAG) is a technique for enhancing the quality of large language model generation. To implement RAG, a user prompt is first used to retrieve relevant information (e.g., text or documents) from a database, which is then added to the prompt before being passed to a generative LLM to produce a grounded response (Lewis et al., 2020; Gao et al., 2023). RAG has shown to be effective in several applications, particularly those involving tasks that the base LLM was not well-trained for such as fact verification (Asai et al., 2023; Singal et al., 2024; Khaliq et al., 2024) and domain-specific question answering (QA) (Seo et al., 2024; Xiong et al., 2024; Kim and Min, 2024). Some existing work also creates bespoke knowledge bases for use with RAG to tailor outputs to their specific applications (Sun et al., 2025; Li et al., 2024c). Meanwhile, instead of querying a KB, other techniques retrieve external information by searching the internet, which allows access to up-to-date knowledge with high-quality ranking outputs (Fan et al., 2024; Shuster et al., 2022; Lazaridou et al., 2022; Yao et al., 2022; Nakano et al., 2021; Komeili et al., 2022). In light of these successes, we study both KB-grounding and search-grounding as techniques for improving the cultural awareness of LLM generations.

2.1 The Knowledge Base Grounding Strategy

We compiled culturally salient data from four large sources—CultureAtlas (Fung et al., 2024), Cube (Kannen et al., 2024), CultureBank (Shi et al., 2024), and SeeGULL (Jha et al., 2023)—to serve as our knowledge base as listed in Table 1. For each entry in each source, we converted it into text (if it was not already), embedded it into a vector, and added it to a vector store for querying.

Fig. 1 (top) shows how the KB-grounding strategy works. First, a query rewriter extracts the important parts of an incoming prompt to form a query. We can configure this step to test different KB queries (e.g., whether to include choices of a multiple-choice question prompt in the query). Next, we use the query to retrieve n documents from the KB. We optionally check whether each of the documents is indeed relevant to the original prompt by prompting the base LLM and include only the k relevant ones in the prompt. We call the process with the relevancy check step *selective RAG*, while *non-selective RAG* includes all n documents in the prompt. Then, we feed the augmented prompt to the LLM to get a raw answer. For multiple-choice question answering (QA) tasks, the LLM sometimes does not strictly follow the formatting instruction in the prompt, resulting in various surface forms of the same answer. For example, the raw outputs for the answer "1) Yes" we found include, e.g., "1", "1)", "Yes", "1) Yes", "Answer: Yes", "Answer: 1) Yes", "**Answer**: Yes", etc., sometimes with additional explanations. Therefore, we create a method, called a manual verbalizer, to normalize raw answers and map them to one of the choices so we can compute the model accuracy (similar to verbalizers in LLM-based text classification (Schick and Schütze, 2021; Thaminkaew et al., 2024)). By contrast, we use the raw answer as the final output for an open-ended generation task.

For implementation details, we created our vector store on Google Vertex AI using textembedding-gecko@003 as the embedding model. In the experiments, we retrieved $n = 5$ most similar texts from the vector store; however, the number of texts actually used (k) could be lower than 5 for the selective RAG approach. In Section 3, we applied KB-grounding to three LLMs: gemini-flash-1.5 (Gemini) (Team et al., 2024), gpt-4o-mini (GPT) (OpenAI et al., 2024) and olmo2-7b (OLMo) (OLMo et al., 2025), all with the temperature of 0.5. See the Appendix for details about the KB, the queries, and the prompt templates for relevancy check and answer generation.

2.2 The Search Grounding Strategy

Fig 1 (bottom) outlines how the search-grounding strategy works. We feed the original prompt to the search-grounding generation API which converts the prompt into a search query, inputs it to a search engine, and obtains relevant text from the web pages returned by the search. Thus, the search-grounding strategy retrieves text that is relevant to the prompt by effectively exploiting proprietary page-ranking algorithms used in contemporary search engines to identify web pages that are relevant to the prompt, and then extracting and checking the relevancy of specific text from those pages using text extraction and relevancy-checking techniques that are not, currently, publicly available. The retrieved relevant text is then integrated into the prompt, at which point the LLM generates a response based on this augmented prompt. Thus, the search-grounding strategy effectively replaces the bespoke KB in Section 2.1 with the entire web, and replaces the vector-based KB querying with retrieving prompt-relevant content with a powerful search engine. We implemented search grounding on Google Vertex AI. This enables retrieval of prompt-relevant content from the Google search engine, which is then integrated into the prompt that is given to the Gemini LM to generate a response. Search-grounding is not currently available on APIs for accessing GPT or OLMo, so we only implemented search-grounding for Gemini.

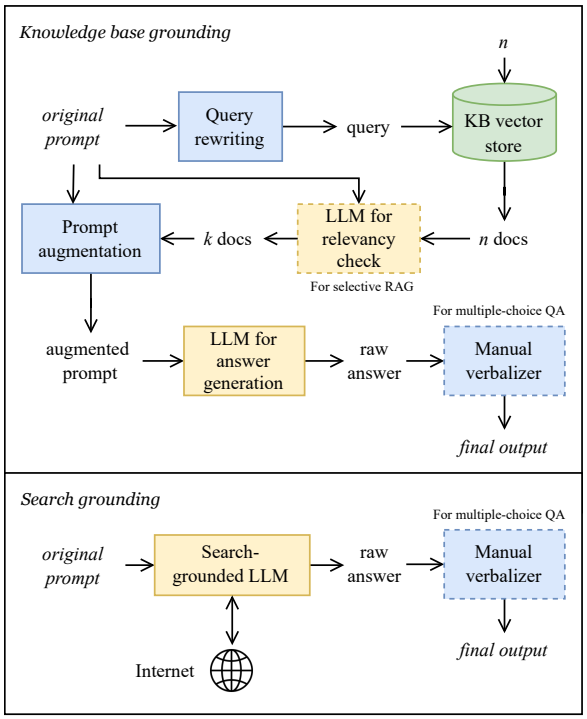


Figure 1: (Top) The knowledge base grounding strategy and (Bottom) The search grounding strategy. Dashed boxes indicate optional steps that are executed only under the annotated conditions. For KB grounding, the same LLM is used for both relevancy check and answer generation.

3 Cultural Competence Benchmarks

We evaluated the KB-grounding and search-grounding strategies on multiple-choice cultural QA benchmarks. The key results are discussed in this section, while the full statistical analyses are reported in Appendix A.5.

3.1 Cultural Knowledge

Setup. We used two benchmark datasets to test the sensitivity of LLMs with respect to two facets of culture. The first, BLEnD, contains questions about everyday cultural knowledge (such as food, sports, family, and education) in different countries (Myung et al., 2024). We used the ~24k English questions from BLEnD that are related to ten countries[2] represented in our bespoke KB. The second, NORMAD, focuses on cultural norms and values (Rao et al., 2024). Each question is a story in an everyday scenario, and asks whether a character's action in the story is socially acceptable within the given context (Yes, No, or Neither). We experimented with two types of contexts – *Country* and *Country+Value*. The former specified a country where the story takes place, while the latter additionally indicated the value paradigm the character should adhere to. NORMAD has ~2.6k questions in total.

[2]China, Ethiopia, Greece, Indonesia, Iran, Mexico, South Korea, Spain, the United Kingdom, and the United States.

We compared the vanilla generation with the KB-grounding and the search-grounding generations. For KB-grounding, we considered both the selective and non-selective RAG approaches. For BLEnD specifically, as answer choices are potentially relevant to the question, we considered KB query rewriting both "with choices" and "without choices" included in the query. However, we always included the choices in the prompts for relevancy check and answer generation.

Results. Fig. 2 (left) presents the results of the BLEnD benchmark. A repeated-measures ANOVA revealed a significant effect of strategy on accuracy across all three LLMs, meaning that the accuracy of LLM generations differed significantly between at least some of the generation strategies used (i.e., search-grounding, the varieties of KB-grounding, or vanilla). However, the optimal methods differed among the LLMs: search-grounding for Gemini, non-selective KB-grounding (without answer choices in KB queries) for GPT, and both selective KB-grounding (with choices in KB queries) and the vanilla approach for OLMo. Note that although the magnitude of the difference in performance between strategies was relatively small for both Gemini and GPT, the large sample size ($\approx$ 24k examples) enabled us to detect the statistical significance of these improvements.

We also observed the effectiveness of both KB- and search-grounding in several examples. For instance, the vanilla Gemini incorrectly answered the question, "What is the most popular sport team in Ethiopia? (A) coffee (B) lg twins (C) persepolis (D) real madrid." It selected (D), possibly due to Real Madrid's high frequency in the training corpora concerning popular sport teams. In fact, the correct answer is (A), as 'coffee' refers to the Ethiopian Coffee Sport Club. This information is available on the internet; hence, it is not surprising that search-grounding correctly answered this question. Notably, search-grounded Gemini achieves 74.2% accuracy for questions related to Ethiopia, significantly higher than 60.3% of the vanilla baseline and 62.9% of the best KB grounding setting. Certain KB-grounding settings also answered the above question correctly, even though 'Ethiopian Coffee' was not mentioned in any of the retrieved texts. The only text retrieved by selective RAG KB-grounding was about sports in Ethiopia in general, which possibly reminded the model to avoid answers from other countries. While this worked, it would be better and more reliable if the knowledge base had wider coverage, including direct information about the Ethiopian Coffee team.

Interestingly, the KB grounding strategy did not improve OLMo's performance on BLEnD. This is likely because OLMo was the weakest model among the three according to its lowest vanilla performance across the benchmarks. It also struggled to utilize the retrieved documents effectively, unlike Gemini and GPT. This issue was prominent when we used the non-selective KB grounding strategy, where the relevancy check step

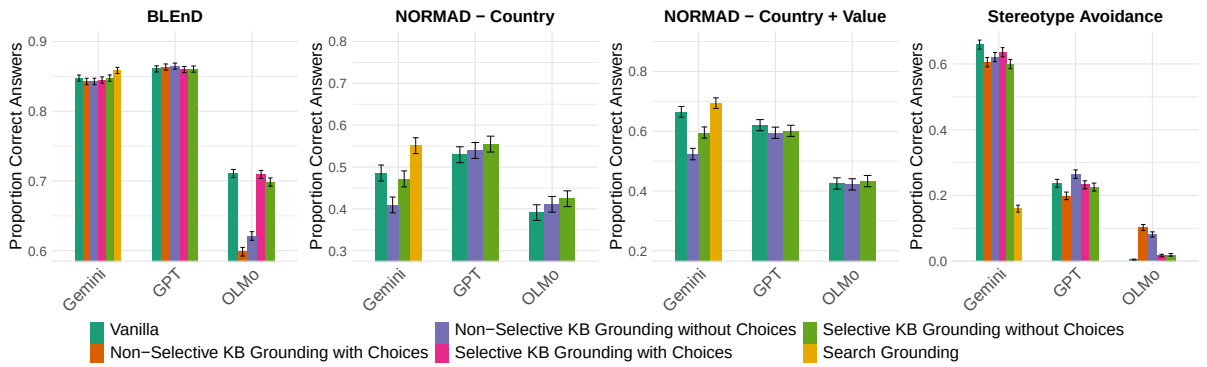


Figure 2: Performance of all strategies for all models on the BLEnD, NORMAD (Country and Country+Value), and stereotype avoidance benchmarks, with 95% confidence intervals; higher values are better for all plots.

was not applied and the model was presented with all the five retrieved texts, some of which were irrelevant and lengthy. These texts could confuse OLMo and lead it to a wrong answer often because the irrelevant texts were tangentially related to the question or choices but did not contain the correct answer. Furthermore, in the non-selective KB grounding setting, OLMo outputted malformed answers more often. BLEnD requires the answer in the JSON format. Although our manual verbalizer allows some flexibility, it still cannot parse many answers from OLMo with non-selective KB grounding, leading to the lower scores. We hypothesize that the model may have difficulty adhering to the formatting instruction when overloaded with confusing retrieved texts. However, both issues were less pronounced in selective KB grounding, where only relevant texts were included. Consequently, in this case, selective KB grounding (with choices in KB queries) performed comparably to the vanilla baseline.

The middle two charts in Fig. 2 present the results for NORMAD for both Country and County+Value context types. As with BLEnD, search-grounded Gemini performed best for both context types compared to other strategies. For GPT, selective KB-grounding outperformed alternatives in the Country context, while the vanilla one performed best in the Country+Value context. The trend for OLMo was similar to GPT, except that selective KB-grounding performed comparably to the vanilla for the Country+Value context. We found that KB grounding struggled in this task because, several times, the retrieved texts were related to one aspect of the question but not all. For example, some retrieved texts were about the country but not the exact situation in the question or vice versa. This could cause the LLMs to give wrong answers. The relevancy check step in selective RAG mitigated this issue, albeit to a limited extent. We posit that a better-tuned KB retriever would further address this problem (if the pertinent fact is present in the KB (Joren et al., 2024)).

3.2 Stereotype Avoidance

Setup. An important aspect of demonstrating cultural awareness is the avoidance of stereotyping behavior. Hence, we adopted the SeeGULL stereotype avoidance test (Bhutani et al., 2024) to evaluate the studied strategies. Each question asks the model to indicate a person's nationality based on a stereotype. For example,

"I know a person who is *a fighter*. This person is: (A) Albanian (B) Afghan (C) Filipino (D) Guyanese. Write your answer on the next line, or write (?) if you are unsure."

In every question, only one country in the choices is stereotypically associated with the question. However, we considered only '(?)' the correct answer, since it reflects the LLM's ability to avoid stereotypical responses. The benchmark contains 4.6k questions. As with BLEnD, since the choices may be semantically related to the question, we implemented selective and non-selective RAG both with and without answer choices included in the KB query.

Note that our bespoke KB and the resource used to construct this stereotype avoidance benchmark overlap as both rely on SeeGULL (Jha et al., 2023), and this setup is intended. Unlike cultural knowledge evaluation in Section 3.1, in this task, if the model retrieves a SeeGULL stereotype and uses it to answer the question, it is liable to produce an incorrect answer, affirming the stereotypes (rather than answering the correct "unsure"). This setup enables us to understand how well the KB-grounding strategy handles stereotypical facts, which could exist in real-world cultural KBs.

Results. Fig. 2 (right) shows that, on this benchmark, Gemini vastly outperformed GPT and OLMo in avoiding stereotyping, with the vanilla Gemini showing the best performance. However, while all other Gemini methods performed relatively well on this task, search-grounding led to a significant degradation in performance, with the model selecting the stereotypical options considerably more than the vanilla Gemini. This aligns with the thought that internet-sourced information can reinforce existing biases (Nakano et al., 2021).

In contrast to BLEnD, the non-selective KB grounding strategy significantly improved OLMo's performance in this task because, when presented with several texts that seem irrelevant to the question, OLMo

could not ground any choice with the retrieved texts and therefore answered "unsure", which was considered a correct answer for this task. By contrast, Gemini may be able to ignore irrelevant texts and exploit the relevant ones, resulting in stereotypical answers.

To understand how stereotypical facts in the KB affected KB grounding, we examined the sources of the retrieved texts and found that, out of 4,600 questions, in the KB query "without choices" setting, 1,156 questions retrieved at least one SeeGULL stereotype. However, in only 35 of these did the stereotype's content exactly match the question. Similarly, in the KB query "with choices" setting, 1,266 questions retrieved at least one SeeGULL stereotype, but the stereotype's content exactly matched the question in only 2 cases. In both settings, SeeGULL stereotypes whose content matched the question always passed the relevancy check for GPT and OLMo and usually passed for Gemini ($\sim$80% of the time). The inclusion of these stereotypes in the prompt flipped the original "unsure" answer of the vanilla model to a stereotypical answer. This suggests that including stereotypes in a prompt can induce a model to affirm stereotypes. However, as the KB has significantly fewer and narrower stereotypical contents than the internet, this issue is not as severe as the search-grounding strategy.

4 Human Evaluation

Setup. To evaluate the strategies on a more open-ended text generation task, we translated five questions from BLEnD and five questions from NORMAD into open-ended prompts asking the model to tell a story set in a particular country. We adapted each of these ten questions for the ten national cultures used in the BLEnD benchmark evaluation, leaving us with 100 (country, prompt) pairs. Next, we generated responses from Gemini using the vanilla, the selective KB-grounding, and the search-grounding. For each strategy, we generated three unique responses with the temperature of 0.5. Then we recruited nine evaluators from each of the ten studied cultures to rate, on a scale from 0 to 4, how culturally familiar each response was and to provide a brief justification of their score (see Appendices A.6–A.7 for details).

Results. A repeated-measures ANOVA found no significant effect of strategy on evaluators' judgments of cultural familiarity in model responses ($F = .18$, $p = .827$) Also, the interaction effect between an evaluator's national culture and the generation strategy used was not significant ($F = .84$, $p = .651$). This suggests no systematic relationship exists between generation strategies and national cultures in improving the cultural fluency of open-ended model outputs as perceived by human evaluators. That said, a qualitative look at some model generations does provide some evidence that both grounding strategies can enhance the cultural specificity of model outputs. In response to the prompt 'Tell me a story in Mexico in which a group of people of varying ages eat together and all guests behave in a socially acceptable way,' the selective KB-grounded and search-grounded responses each mentioned many specific dishes and games, while the vanilla response was far more generic (see Appendix A.8 for examples). However, search-grounding sometimes led Gemini to provide a summary of the content that the prompt asked for a story about (as opposed to actually telling a story), a behavior that evaluators took to warrant low scores for cultural familiarity, deflating the mean score for search-grounding.

5 Discussion and Conclusion

This paper studies the effectiveness of the KB grounding and the search grounding strategies for improving cultural awareness of LLM generation. We discuss the key findings and suggestions below.

KB Grounding vs Search Grounding. The advantages of search-grounding on BLEnD and NORMAD speak to the vast space of cultural facts on the internet. Even the large KB we compiled here still lacked many culturally relevant facts. We also observed some bias in each knowledge source used: approximately 19% of CultureAtlas entries and 25% of CultureBank entries concern the culture of the United States. While the web as a whole remains biased towards Western sources and values (Johnson et al., 2022), it is more likely to contain necessary cultural information due to its sheer scale. However, the poor performance of search-grounding on the stereotype avoidance benchmark reminds us that the context retrieved via web search could reinforce the (typically false) notion that stereotypes are factual and encourage models to affirm those stereotypes. This suggests that search-grounding is not yet a panacea for improving the cultural sensitivity of LLMs.

Knowledge vs Fluency. The results of the human evaluation do not show that search-grounding or KB-grounding improved the cultural familiarity of LLM outputs. The divergence in the performance of both strategies on the human evaluation and the multiple-choice QA benchmarks suggests that we ought to draw a distinction between two varieties of cultural awareness. On the one hand, there is a sense of cultural awareness that involves possessing **propositional knowledge** about a culture (i.e., knowing facts about that culture), as measured by the multiple-choice QA benchmarks, where the grounding strategies can improve the model performance. However, another sense of cultural awareness involves writing and speaking like someone with first-hand experience of and immersion in a culture, i.e., a sense of **cultural fluency**, as measured by our human evaluation, which revealed that the two grounding strategies were of limited value. We leave it to future work to develop strategies for improving the cultural awareness of generative language models along this second axis.

Limitations

We acknowledge the several limitations of our work in this paper. First, we ran our evaluations only on smaller versions of the GPT-4, Gemini 1.5, and OLMO 2 models. It remains an open question whether our pattern of results would be similar for larger versions of these models and other model families.

Second, for BLEnD and the human evaluation, we ran our evaluations using prompts that were relevant to ten national countries and cultures; a more comprehensive study would require the use of a wider range of national *and* regional cultures. With the momentum in the community to create more culturally salient resources, a more comprehensive study in the future will help identify gaps and interventions for the majority world.

Third, we only implement the search-grounding strategy using the Gemini model, specifically using the "Grounding with Google Search" feature of Vertex AI. According to the metadata returned, this end-to-end API has its own methods for query rewriting, incorporating retrieved texts, and providing citations, the details of which are not publicly available. Since these steps are tied to Gemini, it is not replicable for other LLMs while maintaining a fair comparison. As more LLM APIs introduce search-grounding capabilities, we hope that the lessons learned from search-grounding Gemini, as presented in this paper, establish the necessity of carefully auditing the characteristics of any future search-grounded LLMs before using them in culturally sensitive applications. Moreover, the paper retains its core message that RAG (including search-grounding) can improve performance with respect to cultural propositional knowledge, but not necessarily cultural fluency.

Finally, all of our evaluations concerned solely English-language prompts and outputs. While we take it to be an important goal for generative language models that they be able to produce culturally-aware outputs about any culture in the world's most widely-spoken language, the landscape of cultural awareness becomes much more nuanced when one considers the rich variegation that exists in phrasing and dialect across a wide range of languages. We leave it to future work to examine whether the strategies used here can be adapted to a multi-lingual context.

Ethical Considerations

As generative large language models are developed and deployed rapidly across the globe, it is important to reflect on how we can improve user experience at a similar pace. The promise of model utility for a myriad of tasks such as that of a writing assistant, remains unfulfilled if the model is not beneficial or usable for a vast majority. With this work, we attempted to begin adaptation of techniques in NLP to further the cause of cultural awareness and relevance in models. As noted in our limitations, with more comprehensive work across a greater number of cultures and countries, we hope that development of more culturally-aware models will be possible.

References

Muhammad Farid Adilazuarda, Sagnik Mukherjee, Pradhyumna Lavania, Siddhant Shivdutt Singh, Alham Fikri Aji, Jacki O'Neill, Ashutosh Modi, and Monojit Choudhury. 2024. Towards measuring and modeling "culture" in LLMs: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15763–15784, Miami, Florida, USA. Association for Computational Linguistics.

Dhruv Agarwal, Mor Naaman, and Aditya Vashistha. 2025. Ai suggestions homogenize writing toward western styles and diminish cultural nuances.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Mohammad Atari, Mona J Xue, Peter S Park, Damián Blasi, and Joseph Henrich. 2023. Which humans? *PsyArXiv*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Mukul Bhutani, Kevin Robinson, Vinodkumar Prabhakaran, Shachi Dave, and Sunipa Dev. 2024. SeeGULL multilingual: a dataset of geo-culturally situated stereotypes. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 842–854, Bangkok, Thailand. Association for Computational Linguistics.

Alex J Chan, José Luis Redondo García, Fabrizio Silvestri, Colm O'Donnell, and Konstantina Palla. 2023. Enhancing content moderation with culturally-aware models. *arXiv e-prints*, pages arXiv–2312.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models.

In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.

Yi Fung, Ruining Zhao, Jae Doo, Chenkai Sun, and Heng Ji. 2024. Massively multi-cultural knowledge acquisition & lm benchmarking. *arXiv preprint arXiv:2402.09369*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Akshita Jha, Aida Mostafazadeh Davani, Chandan K Reddy, Shachi Dave, Vinodkumar Prabhakaran, and Sunipa Dev. 2023. SeeGULL: A stereotype benchmark with broad geo-cultural coverage leveraging generative models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9851–9870, Toronto, Canada. Association for Computational Linguistics.

Rebecca L Johnson, Giada Pistilli, Natalia Menédez-González, Leslye Denisse Dias Duran, Enrico Panai, Julija Kalpokiene, and Donald Jay Bertulfo. 2022. The ghost in the machine has an american accent: value conflict in gpt-3. *arXiv preprint arXiv:2203.07785*.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. 2024. Sufficient context: A new lens on retrieval augmented generation systems. *arXiv preprint arXiv:2411.06037*.

Nithish Kannen, Arif Ahmad, marco Andreetto, Vinodkumar Prabhakaran, Utsav Prabhu, Adji Bousso Dieng, Pushpak Bhattacharyya, and Shachi Dave. 2024. Beyond aesthetics: Cultural competence in text-to-image models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Mohammed Abdul Khaliq, Paul Yu-Chun Chang, Mingyang Ma, Bernhard Pflugfelder, and Filip Miletić. 2024. RAGAR, your falsehood radar: RAG-augmented reasoning for political fact-checking using multimodal large language models. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 280–296, Miami, Florida, USA. Association for Computational Linguistics.

Jaewoong Kim and Moohong Min. 2024. From rag to qa-rag: Integrating generative ai for pharmaceutical regulatory compliance process. *arXiv preprint arXiv:2402.01717*.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8460–8478, Dublin, Ireland. Association for Computational Linguistics.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Cheng Li, Mengzhou Chen, Jindong Wang, Sunayana Sitaram, and Xing Xie. 2024a. Culturellm: Incorporating cultural differences into large language models. In *Thirty-Eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Cheng Li, Damien Teney, Linyi Yang, Qingsong Wen, Xing Xie, and Jindong Wang. 2024b. Culturepark: Boosting cross-cultural understanding in large language models. In *Thirty-Eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Jiarui Li, Ye Yuan, and Zehua Zhang. 2024c. Enhancing llm factual accuracy with rag to counter hallucinations: A case study on domain-specific queries in private knowledge-bases. *arXiv preprint arXiv:2403.10446*.

Junho Myung, Nayeon Lee, Yi Zhou, Jiho Jin, Rifki Afina Putri, Dimosthenis Antypas, Hsuvas Borkakoty, Eunsu Kim, Carla Perez-Almendros, Abinew Ali Ayele, Víctor Gutiérrez-Basulto, Yazmín Ibáñez-García, Hwaran Lee, Shamsuddeen Hassan Muhammad, Kiwoong Park, Anar Sabuhi Rzayev, Nina White, Seid Muhie Yimam, Mohammad Taher Pilehvar, Nedjma Ousidhoum, Jose Camacho-Collados, and Alice Oh. 2024. Blend: A benchmark for llms on everyday knowledge in diverse cultures and languages.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. 2 olmo 2 furious.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen

He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024. Gpt-4o system card.

Siddhesh Pawar, Junyeong Park, Jiho Jin, Arnav Arora, Junho Myung, Srishti Yadav, Faiz Ghifari Haznitrama, Inhwa Song, Alice Oh, and Isabelle Augenstein. 2024. Survey of cultural awareness in language models: Text and beyond.

Vinodkumar Prabhakaran, Rida Qadri, and Ben Hutchinson. 2022. Cultural incongruencies in artificial intelligence. *arXiv preprint arXiv:2211.13069*.

Rida Qadri, Aida M. Davani, Kevin Robinson, and Vinodkumar Prabhakaran. 2025. Risks of cultural erasure in large language models.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Abhinav Rao, Akhila Yerukola, Vishwa Shah, Katharina Reinecke, and Maarten Sap. 2024. Normad: A framework for measuring the cultural adaptability of large language models.

Abhinav Sukumar Rao, Aditi Khandelwal, Kumar Tanmay, Utkarsh Agarwal, and Monojit Choudhury. 2023. Ethical reasoning over moral alignment: A case and framework for in-context ethical policies in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13370–13388, Singapore. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Minju Seo, Jinheon Baek, James Thorne, and Sung Ju Hwang. 2024. Retrieval-augmented data augmentation for low-resource domain tasks. *arXiv preprint arXiv:2402.13482*.

Weiyan Shi, Ryan Li, Yutong Zhang, Caleb Ziems, Raya Horesh, Rogério Abreu de Paula, Diyi Yang, et al. 2024. Culturebank: An online community-driven knowledge base towards culturally aware language technologies. *arXiv preprint arXiv:2404.15238*.

Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. 2022. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*.

Ronit Singal, Pransh Patwa, Parth Patwa, Aman Chadha, and Amitava Das. 2024. Evidence-backed fact checking using RAG and few-shot in-context learning with LLMs. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 91–98, Miami, Florida, USA. Association for Computational Linguistics.

Binhuan Sun, Liubov Pashkova, Pascal Aldo Pieters, Archana Sanjay Harke, Omkar Satyavan Mohite, Alberto Santos, Daniel C Zielinski, Bernhard O Palsson, and Patrick Victor Phaneuf. 2025. Pankb: An interactive microbial pangenome knowledgebase for research, biotechnological innovation, and knowledge mining. *Nucleic Acids Research*, 53(D1):D806–D818.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Sercinoglu, Zach Gleicher, Juliette Love, Paul Voigtlaender, Rohan Jain, Gabriela Surita, Kareem Mohamed, Rory Blevins, Junwhan Ahn, Tao Zhu, Kornraphop Kawintiranon, Orhan Firat, Yiming Gu, Yujing Zhang, Matthew Rahtz, Manaal Faruqui, Natalie Clay, Justin Gilmer, JD Co-Reyes, Ivo Penchev, Rui Zhu, Nobuyuki Morioka, Kevin Hui, Krishna Haridasan, Victor Campos, Mahdis Mahdieh, Mandy Guo, Samer Hassan, Kevin Kilgour, Arpi Vezer, Heng-Tze Cheng, Raoul de Liedekerke, Siddharth Goyal, Paul Barham, DJ Strouse, Seb Noury, Jonas Adler, Mukund Sundararajan, Sharad Vikram, Dmitry Lepikhin, Michela Paganini, Xavier Garcia, Fan Yang, Dasha Valter, Maja Trebacz, Kiran Vodrahalli, Chulayuth Asawaroengchai, Roman Ring, Norbert Kalb, Livio Baldini Soares, Siddhartha Brahma, David Steiner, Tianhe Yu, Fabian Mentzer, Antoine He, Lucas Gonzalez, Bibo Xu, Raphael Lopez Kaufman, Laurent El Shafey, Junhyuk Oh, Tom Hennigan, George van den Driessche, Seth Odoom, Mario Lucic, Becca Roelofs, Sid Lall, Amit Marathe, Betty Chan, Santiago Ontanon, Luheng He, Denis Teplyashin, Jonathan Lai, Phil Crone, Bogdan Damoc, Lewis Ho, Sebastian Riedel, Karel Lenc, Chih-Kuan Yeh, Aakanksha Chowdhery, Yang Xu, Mehran Kazemi, Ehsan Amid, Anastasia Petrushkina, Kevin Swersky, Ali Khodaei, Gowoon Chen, Chris Larkin, Mario Pinto, Geng Yan, Adria Puigdomenech Badia, Piyush Patil, Steven Hansen, Dave Orr, Sebastien M. R. Arnold, Jordan Grimstad, Andrew Dai, Sholto Douglas, Rishika Sinha, Vikas Yadav, Xi Chen, Elena Gribovskaya, Jacob Austin, Jeffrey Zhao, Kaushal Patel, Paul Komarek, Sophia Austin, Sebastian Borgeaud, Linda Friso, Abhimanyu Goyal, Ben Caine, Kris Cao, Da-Woon Chung, Matthew Lamm, Gabe Barth-Maron, Thais Kagohara, Kate Olszewska, Mia Chen, Kaushik Shivakumar, Rishabh Agarwal, Harshal Godhia, Ravi Rajwar, Javier Snaider, Xerxes Dotiwalla, Yuan Liu, Aditya Barua, Victor Ungureanu, Yuan Zhang, Bat-Orgil Batsaikhan, Mateo Wirth, James Qin, Ivo Danihelka, Tulsee Doshi, Martin Chadwick, Jilin Chen, Sanil Jain, Quoc Le, Arjun Kar, Madhu Gurumurthy,

Cheng Li, Ruoxin Sang, Fangyu Liu, Lampros Lamprou, Rich Munoz, Nathan Lintz, Harsh Mehta, Heidi Howard, Malcolm Reynolds, Lora Aroyo, Quan Wang, Lorenzo Blanco, Albin Cassirer, Jordan Griffith, Dipanjan Das, Stephan Lee, Jakub Sygnowski, Zach Fisher, James Besley, Richard Powell, Zafarali Ahmed, Dominik Paulus, David Reitter, Zalan Borsos, Rishabh Joshi, Aedan Pope, Steven Hand, Vittorio Selo, Vihan Jain, Nikhil Sethi, Megha Goel, Takaki Makino, Rhys May, Zhen Yang, Johan Schalkwyk, Christina Butterfield, Anja Hauth, Alex Goldin, Will Hawkins, Evan Senter, Sergey Brin, Oliver Woodman, Marvin Ritter, Eric Noland, Minh Giang, Vijay Bolina, Lisa Lee, Tim Blyth, Ian Mackinnon, Machel Reid, Obaid Sarvana, David Silver, Alexander Chen, Lily Wang, Loren Maggiore, Oscar Chang, Nithya Attaluri, Gregory Thornton, Chung-Cheng Chiu, Oskar Bunyan, Nir Levine, Timothy Chung, Evgenii Eltyshev, Xiance Si, Timothy Lillicrap, Demetra Brady, Vaibhav Aggarwal, Boxi Wu, Yuanzhong Xu, Ross McIlroy, Kartikeya Badola, Paramjit Sandhu, Erica Moreira, Wojciech Stokowiec, Ross Hemsley, Dong Li, Alex Tudor, Pranav Shyam, Elahe Rahimtoroghi, Salem Haykal, Pablo Sprechmann, Xiang Zhou, Diana Mincu, Yujia Li, Ravi Addanki, Kalpesh Krishna, Xiao Wu, Alexandre Frechette, Matan Eyal, Allan Dafoe, Dave Lacey, Jay Whang, Thi Avrahami, Ye Zhang, Emanuel Taropa, Hanzhao Lin, Daniel Toyama, Eliza Rutherford, Motoki Sano, HyunJeong Choe, Alex Tomala, Chalence Safranek-Shrader, Nora Kassner, Mantas Pajarskas, Matt Harvey, Sean Sechrist, Meire Fortunato, Christina Lyu, Gamaleldin Elsayed, Chenkai Kuang, James Lottes, Eric Chu, Chao Jia, Chih-Wei Chen, Peter Humphreys, Kate Baumli, Connie Tao, Rajkumar Samuel, Cicero Nogueira dos Santos, Anders Andreassen, Nemanja Rakićević, Dominik Grewe, Aviral Kumar, Stephanie Winkler, Jonathan Caton, Andrew Brock, Sid Dalmia, Hannah Sheahan, Iain Barr, Yingjie Miao, Paul Natsev, Jacob Devlin, Feryal Behbahani, Flavien Prost, Yanhua Sun, Artiom Myaskovsky, Thanumalayan Sankaranarayana Pillai, Dan Hurt, Angeliki Lazaridou, Xi Xiong, Ce Zheng, Fabio Pardo, Xiaowei Li, Dan Horgan, Joe Stanton, Moran Ambar, Fei Xia, Alejandro Lince, Mingqiu Wang, Basil Mustafa, Albert Webson, Hyo Lee, Rohan Anil, Martin Wicke, Timothy Dozat, Abhishek Sinha, Enrique Piqueras, Elahe Dabir, Shyam Upadhyay, Anudhyan Boral, Lisa Anne Hendricks, Corey Fry, Josip Djolonga, Yi Su, Jake Walker, Jane Labanowski, Ronny Huang, Vedant Misra, Jeremy Chen, RJ Skerry-Ryan, Avi Singh, Shruti Rijhwani, Dian Yu, Alex Castro-Ros, Beer Changpinyo, Romina Datta, Sumit Bagri, Arnar Mar Hrafnkelsson, Marcello Maggioni, Daniel Zheng, Yury Sulsky, Shaobo Hou, Tom Le Paine, Antoine Yang, Jason Riesa, Dominika Rogozinska, Dror Marcus, Dalia El Badawy, Qiao Zhang, Luyu Wang, Helen Miller, Jeremy Greer, Lars Lowe Sjos, Azade Nova, Heiga Zen, Rahma Chaabouni, Mihaela Rosca, Jiepu Jiang, Charlie Chen, Ruibo Liu, Tara Sainath, Maxim Krikun, Alex Polozov, Jean-Baptiste Lespiau, Josh Newlan, Zeyncep Cankara, Soo Kwak, Yunhan Xu, Phil Chen, Andy Coenen, Clemens Meyer, Katerina Tsihlas, Ada Ma, Juraj Gottweis, Jinwei Xing, Chenjie Gu, Jin Miao, Christian Frank, Zeynep Cankara, Sanjay Ganapathy, Ishita Dasgupta, Steph Hughes-Fitt, Heng Chen, David Reid, Keran Rong, Hongmin Fan, Joost van Amersfoort, Vincent Zhuang, Aaron Cohen, Shixiang Shane Gu, Anhad Mohananey, Anastasija Ilic, Taylor Tobin, John Wieting, Anna Bortsova, Phoebe Thacker, Emma Wang, Emily Caveness, Justin Chiu, Eren Sezener, Alex Kaskasoli, Steven Baker, Katie Millican, Mohamed Elhawaty, Kostas Aisopos, Carl Lebsack, Nathan Byrd, Hanjun Dai, Wenhao Jia, Matthew Wiethoff, Elnaz Davoodi, Albert Weston, Lakshman Yagati, Arun Ahuja, Isabel Gao, Golan Pundak, Susan Zhang, Michael Azzam, Khe Chai Sim, Sergi Caelles, James Keeling, Abhanshu Sharma, Andy Swing, YaGuang Li, Chenxi Liu, Carrie Grimes Bostock, Yamini Bansal, Zachary Nado, Ankesh Anand, Josh Lipschultz, Abhijit Karmarkar, Lev Proleev, Abe Ittycheriah, Soheil Hassas Yeganeh, George Polovets, Aleksandra Faust, Jiao Sun, Alban Rrustemi, Pen Li, Rakesh Shivanna, Jeremiah Liu, Chris Welty, Federico Lebron, Anirudh Baddepudi, Sebastian Krause, Emilio Parisotto, Radu Soricut, Zheng Xu, Dawn Bloxwich, Melvin Johnson, Behnam Neyshabur, Justin Mao-Jones, Renshen Wang, Vinay Ramasesh, Zaheer Abbas, Arthur Guez, Constant Segal, Duc Dung Nguyen, James Svensson, Le Hou, Sarah York, Kieran Milan, Sophie Bridgers, Wiktor Gworek, Marco Tagliasacchi, James Lee-Thorp, Michael Chang, Alexey Guseynov, Ale Jakse Hartman, Michael Kwong, Ruizhe Zhao, Sheleem Kashem, Elizabeth Cole, Antoine Miech, Richard Tanburn, Mary Phuong, Filip Pavetic, Sebastien Cevey, Ramona Comanescu, Richard Ives, Sherry Yang, Cosmo Du, Bo Li, Zizhao Zhang, Mariko Iinuma, Clara Huiyi Hu, Aurko Roy, Shaan Bijwadia, Zhenkai Zhu, Danilo Martins, Rachel Saputro, Anita Gergely, Steven Zheng, Dawei Jia, Ioannis Antonoglou, Adam Sadovsky, Shane Gu, Yingying Bi, Alek Andreev, Sina Samangooei, Mina Khan, Tomas Kocisky, Angelos Filos, Chintu Kumar, Colton Bishop, Adams Yu, Sarah Hodkinson, Sid Mittal, Premal Shah, Alexandre Moufarek, Yong Cheng, Adam Bloniarz, Jaehoon Lee, Pedram Pejman, Paul Michel, Stephen Spencer, Vladimir Feinberg, Xuehan Xiong, Nikolay Savinov, Charlotte Smith, Siamak Shakeri, Dustin Tran, Mary Chesus, Bernd Bohnet, George Tucker, Tamara von Glehn, Carrie Muir, Yiran Mao, Hideto Kazawa, Ambrose Slone, Kedar Soparkar, Disha Shrivastava, James Cobon-Kerr, Michael Sharman, Jay Pavagadhi, Carlos Araya, Karolis Misiunas, Nimesh Ghelani, Michael Laskin, David Barker, Qiujia Li, Anton Briukhov, Neil Houlsby, Mia Glaese, Balaji Lakshminarayanan, Nathan Schucher, Yunhao Tang, Eli Collins, Hyeontaek Lim, Fangxiaoyu Feng, Adria Recasens, Guangda Lai, Alberto Magni, Nicola De Cao, Aditya Siddhant, Zoe Ashwood, Jordi Orbay, Mostafa Dehghani, Jenny Brennan, Yifan He, Kelvin Xu, Yang Gao, Carl Saroufim, James Molloy, Xinyi Wu, Seb Arnold, Solomon

Chang, Julian Schrittwieser, Elena Buchatskaya, Soroush Radpour, Martin Polacek, Skye Giordano, Ankur Bapna, Simon Tokumine, Vincent Hellendoorn, Thibault Sottiaux, Sarah Cogan, Aliaksei Severyn, Mohammad Saleh, Shantanu Thakoor, Laurent Shefey, Siyuan Qiao, Meenu Gaba, Shuo yiin Chang, Craig Swanson, Biao Zhang, Benjamin Lee, Paul Kishan Rubenstein, Gan Song, Tom Kwiatkowski, Anna Koop, Ajay Kannan, David Kao, Parker Schuh, Axel Stjerngren, Golnaz Ghiasi, Gena Gibson, Luke Vilnis, Ye Yuan, Felipe Tiengo Ferreira, Aishwarya Kamath, Ted Klimenko, Ken Franko, Kefan Xiao, Indro Bhattacharya, Miteyan Patel, Rui Wang, Alex Morris, Robin Strudel, Vivek Sharma, Peter Choy, Sayed Hadi Hashemi, Jessica Landon, Mara Finkelstein, Priya Jhakra, Justin Frye, Megan Barnes, Matthew Mauger, Dennis Daun, Khuslen Baatarsukh, Matthew Tung, Wael Farhan, Henryk Michalewski, Fabio Viola, Felix de Chaumont Quitry, Charline Le Lan, Tom Hudson, Qingze Wang, Felix Fischer, Ivy Zheng, Elspeth White, Anca Dragan, Jean baptiste Alayrac, Eric Ni, Alexander Pritzel, Adam Iwanicki, Michael Isard, Anna Bulanova, Lukas Zilka, Ethan Dyer, Devendra Sachan, Srivatsan Srinivasan, Hannah Muckenhirn, Honglong Cai, Amol Mandhane, Mukarram Tariq, Jack W. Rae, Gary Wang, Kareem Ayoub, Nicholas FitzGerald, Yao Zhao, Woohyun Han, Chris Alberti, Dan Garrette, Kashyap Krishnakumar, Mai Gimenez, Anselm Levskaya, Daniel Sohn, Josip Matak, Inaki Iturrate, Michael B. Chang, Jackie Xiang, Yuan Cao, Nishant Ranka, Geoff Brown, Adrian Hutter, Vahab Mirrokni, Nanxin Chen, Kaisheng Yao, Zoltan Egyed, Francois Galilee, Tyler Liechty, Praveen Kallakuri, Evan Palmer, Sanjay Ghemawat, Jasmine Liu, David Tao, Chloe Thornton, Tim Green, Mimi Jasarevic, Sharon Lin, Victor Cotruta, Yi-Xuan Tan, Noah Fiedel, Hongkun Yu, Ed Chi, Alexander Neitz, Jens Heitkaemper, Anu Sinha, Denny Zhou, Yi Sun, Charbel Kaed, Brice Hulse, Swaroop Mishra, Maria Georgaki, Sneha Kudugunta, Clement Farabet, Izhak Shafran, Daniel Vlasic, Anton Tsitsulin, Rajagopal Ananthanarayanan, Alen Carin, Guolong Su, Pei Sun, Shashank V, Gabriel Carvajal, Josef Broder, Iulia Comsa, Alena Repina, William Wong, Warren Weilun Chen, Peter Hawkins, Egor Filonov, Lucia Loher, Christoph Hirnschall, Weiyi Wang, Jingchen Ye, Andrea Burns, Hardie Cate, Diana Gage Wright, Federico Piccinini, Lei Zhang, Chu-Cheng Lin, Ionel Gog, Yana Kulizhskaya, Ashwin Sreevatsa, Shuang Song, Luis C. Cobo, Anand Iyer, Chetan Tekur, Guillermo Garrido, Zhuyun Xiao, Rupert Kemp, Huaixiu Steven Zheng, Hui Li, Ananth Agarwal, Christel Ngani, Kati Goshvadi, Rebeca Santamaria-Fernandez, Wojciech Fica, Xinyun Chen, Chris Gorgolewski, Sean Sun, Roopal Garg, Xinyu Ye, S. M. Ali Eslami, Nan Hua, Jon Simon, Pratik Joshi, Yelin Kim, Ian Tenney, Sahitya Potluri, Lam Nguyen Thiet, Quan Yuan, Florian Luisier, Alexandra Chronopoulou, Salvatore Scellato, Praveen Srinivasan, Minmin Chen, Vinod Koverkathu, Valentin Dalibard, Yaming Xu, Brennan Saeta, Keith Anderson, Thibault Sellam, Nick Fernando, Fantine Huot, Junehyuk Jung, Mani Varadarajan, Michael Quinn, Amit Raul, Maigo Le, Ruslan Habalov, Jon Clark, Komal Jalan, Kalesha Bullard, Achintya Singhal, Thang Luong, Boyu Wang, Sujeevan Rajayogam, Julian Eisenschlos, Johnson Jia, Daniel Finchelstein, Alex Yakubovich, Daniel Balle, Michael Fink, Sameer Agarwal, Jing Li, Dj Dvijotham, Shalini Pal, Kai Kang, Jaclyn Konzelmann, Jennifer Beattie, Olivier Dousse, Diane Wu, Remi Crocker, Chen Elkind, Siddhartha Reddy Jonnalagadda, Jong Lee, Dan Holtmann-Rice, Krystal Kallarackal, Rosanne Liu, Denis Vnukov, Neera Vats, Luca Invernizzi, Mohsen Jafari, Huanjie Zhou, Lilly Taylor, Jennifer Prendki, Marcus Wu, Tom Eccles, Tianqi Liu, Kavya Kopparapu, Francoise Beaufays, Christof Angermueller, Andreea Marzoca, Shourya Sarcar, Hilal Dib, Jeff Stanway, Frank Perbet, Nejc Trdin, Rachel Sterneck, Andrey Khorlin, Dinghua Li, Xihui Wu, Sonam Goenka, David Madras, Sasha Goldshtein, Willi Gierke, Tong Zhou, Yaxin Liu, Yannie Liang, Anais White, Yunjie Li, Shreya Singh, Sanaz Bahargam, Mark Epstein, Sujoy Basu, Li Lao, Adnan Ozturel, Carl Crous, Alex Zhai, Han Lu, Zora Tung, Neeraj Gaur, Alanna Walton, Lucas Dixon, Ming Zhang, Amir Globerson, Grant Uy, Andrew Bolt, Olivia Wiles, Milad Nasr, Ilia Shumailov, Marco Selvi, Francesco Piccinno, Ricardo Aguilar, Sara McCarthy, Misha Khalman, Mrinal Shukla, Vlado Galic, John Carpenter, Kevin Villela, Haibin Zhang, Harry Richardson, James Martens, Matko Bosnjak, Shreyas Rammohan Belle, Jeff Seibert, Mahmoud Alnahlawi, Brian McWilliams, Sankalp Singh, Annie Louis, Wen Ding, Dan Popovici, Lenin Simicich, Laura Knight, Pulkit Mehta, Nishesh Gupta, Chongyang Shi, Saaber Fatehi, Jovana Mitrovic, Alex Grills, Joseph Pagadora, Tsendsuren Munkhdalai, Dessie Petrova, Danielle Eisenbud, Zhishuai Zhang, Damion Yates, Bhavishya Mittal, Nilesh Tripuraneni, Yannis Assael, Thomas Brovelli, Prateek Jain, Mihajlo Velimirovic, Canfer Akbulut, Jiaqi Mu, Wolfgang Macherey, Ravin Kumar, Jun Xu, Haroon Qureshi, Gheorghe Comanici, Jeremy Wiesner, Zhitao Gong, Anton Ruddock, Matthias Bauer, Nick Felt, Anirudh GP, Anurag Arnab, Dustin Zelle, Jonas Rothfuss, Bill Rosgen, Ashish Shenoy, Bryan Seybold, Xinjian Li, Jayaram Mudigonda, Goker Erdogan, Jiawei Xia, Jiri Simsa, Andrea Michi, Yi Yao, Christopher Yew, Steven Kan, Isaac Caswell, Carey Radebaugh, Andre Elisseeff, Pedro Valenzuela, Kay McKinney, Kim Paterson, Albert Cui, Eri Latorre-Chimoto, Solomon Kim, William Zeng, Ken Durden, Priya Ponnapalli, Tiberiu Sosea, Christopher A. Choquette-Choo, James Manyika, Brona Robenek, Harsha Vashisht, Sebastien Pereira, Hoi Lam, Marko Velic, Denese Owusu-Afriyie, Katherine Lee, Tolga Bolukbasi, Alicia Parrish, Shawn Lu, Jane Park, Balaji Venkatraman, Alice Talbert, Lambert Rosique, Yuchung Cheng, Andrei Sozanschi, Adam Paszke, Praveen Kumar, Jessica Austin, Lu Li, Khalid Salama, Bartek Perz, Wooyeol Kim, Nandita Dukkipati, Anthony Baryshnikov, Christos Kaplanis, XiangHai Sheng, Yuri Chervonyi, Caglar Unlu, Diego de Las Casas, Harry

Askham, Kathryn Tunyasuvunakool, Felix Gimeno, Siim Poder, Chester Kwak, Matt Miecnikowski, Vahab Mirrokni, Alek Dimitriev, Aaron Parisi, Dangyi Liu, Tomy Tsai, Toby Shevlane, Christina Kouridi, Drew Garmon, Adrian Goedeckemeyer, Adam R. Brown, Anitha Vijayakumar, Ali Elqursh, Sadegh Jazayeri, Jin Huang, Sara Mc Carthy, Jay Hoover, Lucy Kim, Sandeep Kumar, Wei Chen, Courtney Biles, Garrett Bingham, Evan Rosen, Lisa Wang, Qijun Tan, David Engel, Francesco Pongetti, Dario de Cesare, Dongseong Hwang, Lily Yu, Jennifer Pullman, Srini Narayanan, Kyle Levin, Siddharth Gopal, Megan Li, Asaf Aharoni, Trieu Trinh, Jessica Lo, Norman Casagrande, Roopali Vij, Loic Matthey, Bramandia Ramadhana, Austin Matthews, CJ Carey, Matthew Johnson, Kremena Goranova, Rohin Shah, Shereen Ashraf, Kingshuk Dasgupta, Rasmus Larsen, Yicheng Wang, Manish Reddy Vuyyuru, Chong Jiang, Joana Ijazi, Kazuki Osawa, Celine Smith, Ramya Sree Boppana, Taylan Bilal, Yuma Koizumi, Ying Xu, Yasemin Altun, Nir Shabat, Ben Bariach, Alex Korchemniy, Kiam Choo, Olaf Ronneberger, Chimezie Iwuanyanwu, Shubin Zhao, David Soergel, Cho-Jui Hsieh, Irene Cai, Shariq Iqbal, Martin Sundermeyer, Zhe Chen, Elie Bursztein, Chaitanya Malaviya, Fadi Biadsy, Prakash Shroff, Inderjit Dhillon, Tejasi Latkar, Chris Dyer, Hannah Forbes, Massimo Nicosia, Vitaly Nikolaev, Somer Greene, Marin Georgiev, Pidong Wang, Nina Martin, Hanie Sedghi, John Zhang, Praseem Banzal, Doug Fritz, Vikram Rao, Xuezhi Wang, Jiageng Zhang, Viorica Patraucean, Dayou Du, Igor Mordatch, Ivan Jurin, Lewis Liu, Ayush Dubey, Abhi Mohan, Janek Nowakowski, Vlad-Doru Ion, Nan Wei, Reiko Tojo, Maria Abi Raad, Drew A. Hudson, Vaishakh Keshava, Shubham Agrawal, Kevin Ramirez, Zhichun Wu, Hoang Nguyen, Ji Liu, Madhavi Sewak, Bryce Petrini, DongHyun Choi, Ivan Philips, Ziyue Wang, Ioana Bica, Ankush Garg, Jarek Wilkiewicz, Priyanka Agrawal, Xiaowei Li, Danhao Guo, Emily Xue, Naseer Shaik, Andrew Leach, Sadh MNM Khan, Julia Wiesinger, Sammy Jerome, Abhishek Chakladar, Alek Wenjiao Wang, Tina Ornduff, Folake Abu, Alireza Ghaffarkhah, Marcus Wainwright, Mario Cortes, Frederick Liu, Joshua Maynez, Andreas Terzis, Pouya Samangouei, Riham Mansour, Tomasz Kępa, François-Xavier Aubet, Anton Algymr, Dan Banica, Agoston Weisz, Andras Orban, Alexandre Senges, Ewa Andrejczuk, Mark Geller, Niccolo Dal Santo, Valentin Anklin, Majd Al Merey, Martin Baeuml, Trevor Strohman, Junwen Bai, Slav Petrov, Yonghui Wu, Demis Hassabis, Koray Kavukcuoglu, Jeff Dean, and Oriol Vinyals. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.

Thanakorn Thaminkaew, Piyawat Lertvittayakumjorn, and Peerapon Vateekul. 2024. Label-aware automatic verbalizer for few-shot text classification in mid-to-low resource languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 101–109, Bangkok, Thailand. Association for Computational Linguistics.

Wenxuan Wang, Wenxiang Jiao, Jingyuan Huang, Ruyi Dai, Jen-tse Huang, Zhaopeng Tu, and Michael Lyu. 2024. Not all countries celebrate thanksgiving: On the cultural dominance in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6349–6384, Bangkok, Thailand. Association for Computational Linguistics.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. *arXiv preprint arXiv:2402.13178*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

A Appendix

A.1 The Bespoke Knowledge Base

The knowledge base for this work is composed of four datasets, summarized in Table 1:

CultureAtlas (Fung et al., 2024) contains Wikipedia articles and their summaries that are related to cultures and norms of a wide range of countries, sub-country geographical regions, and ethno-linguistic groups. Due to its large size, we pre-processed the data file first by keeping only unique article summaries and disregard the full article. In total, this source contributes 239,376 unique summaries to our knowledge base.

Cube (Kannen et al., 2024) contains cultural concepts from three domains (i.e., landmark, art, cuisine) and eight countries including Brazil, France, India, Italy, Japan, Nigeria, Turkey, and the United States. Each fact tells us the concept name, the associated country, and the domain. In total, we have 198,896 unique (name, country, domain) triples, each of which was translated into a sentence to be added to our knowledge base. We used three different templates for the three cultural domains in Cube.

- **Landmark** – "<name> is a place in <country>."
- **Art** – "<name> is an art concept in <country>."
- **Cuisine** – "<name> is from <country> cuisine." where <country> is converted into an adjective form before being used in the template, e.g., France → French and Nigeria → Nigerian.

For example, (Pamonha, Brazil, cuisine) became "Pamonha is from Brazillian cuisine" in our knowledge base, rendering the triple into a natural-language format appropriate for embedding.

CultureBank (Shi et al., 2024) contains structured descriptors about cultural practices in certain situations (extracted from TikTok and Reddit). Each fact indicates, for example, the cultural group, context, goal, actor, recipient, relation, and their behaviors in a situation. In total, we have 22,990 cultural descriptors. Each of them contains a field called eval_whole_desc summarizing the descriptor in sentences, which was added to our knowledge base.

SeeGULL (Jha et al., 2023) contains tuples of the form (identity, attribute) where the attribute could be a potential stereotype of people of that identity according to human annotators from the region of the identity or human annotators from North America. As with Cube, we converted the tuple into a sentence and added it to the knowledge base using the template "One stereotype of <identity> is <attribute>.". In total, there are 6,871 sentences from SeeGULL in our knowledge base.

Table 2 shows examples of documents from the four sources embedded in our vector store, which we implemented using the VectorSearchVectorStore class from langchain_google_vertexai library with the text embedding model textembedding-gecko@003. In the experiments, we always used $n = 5$, i.e., retrieving five documents from the knowledge base for each query.

CultureAtlas (Fung et al., 2024) • The culture of Assam is traditionally a hybrid one, developed due to cultural assimilation of different ethno-cultural groups under various political-economic systems in different periods of its history. • Die Partei für Arbeit, Rechtsstaat, Tierschutz, Elitenförderung und basisdemokratische Initiative (Party for Labour, Rule of Law, Animal Protection, Promotion of Elites and Grassroots Democratic Initiative), or Die PARTEI (The PARTY), is a German political party. It was founded in 2004 by the editors of the German satirical magazine Titanic. It is led by Martin Sonneborn. In the 2014 European Parliament election, the party won a seat, marking the first time that a satirical party has won a seat to the European Parliament. With the 2019 European Parliament election, the party gained a second seat, held by Nico Semsrott.
Cube (Kannen et al., 2024) • Manihot Esculenta is from Brazilian cuisine. • Gangan drumming is an art concept in Nigeria.
CultureBank (Shi et al., 2024) • In the UK, it is common for people to engage in various fruit-related practices, such as importing, growing, and consuming fresh and dried fruit, with a particular preference for tropical varieties. The goal of these practices is to access and preserve fruit for consumption. Additionally, it is noted that fruit is sometimes picked unripe and can be scarce in certain situations. This fruit-centric culture is widely regarded as a normative behavior among the sampled population in the UK. • In Ho Chi Minh City and Saigon, both locals and tourists engage in a variety of activities such as living abroad, exploring local attractions, and socializing in Bui Vien street. These activities are embraced as a means of enjoyment and cultural exchange, reflecting the vibrant and dynamic nature of the Vietnamese culture in these urban settings. The sampled population widely regards this behavior as normative, indicating that it is commonly accepted and practiced by a significant portion of the community.
SeeGULL (Jha et al., 2023) • One stereotype of Japanese is conventional. • One stereotype of Mexican is unintelligent.

Table 2: Examples of documents in our bespoke knowledge base.

A.2 Prompts and Queries for BLEnD

Table 3 shows prompts and queries used for the BLEnD dataset. The original prompt of each question consists of four parts – the question, the instruction, the choices, and the 'Answer:' prompt. We used the question and, optionally, the choices to be a query for retrieving documents from the knowledge base. Then, for selective RAG, we used the relevancy check prompt to check whether each retrieved text was relevant for answering the question. Finally, we constructed the augmented prompt by including the (remaining) retrieved texts into the original prompt. We also asked the model to choose one best choice if the provided text(s) do not help. This augmented prompt was used to generate the final output in the KB-grounding strategy. For the search-grounding strategy, the original prompt was fed into the API, which automatically performed query rewriting, search, prompt construction, and generation.

Original prompt: What is the most popular fruit in the UK? Without any explanation, choose only one from the given alphabet choices(e.g., A, B, C). Provide as JSON format: {"answer_choice":""} A. apple B. durian C. mango D. orange Answer:
Query without choices: What is the most popular fruit in the UK?
Query with choices: What is the most popular fruit in the UK? A. apple B. durian C. mango D. orange
Relevancy check prompt: Task: You will be given a question and a piece of information. Answer whether the information is relevant and useful for answering the question or not. Question: "What is the most popular fruit in the UK? A. apple B. durian C. mango D. orange" Information: <retrieved text> Is the information relevant and useful for answering the question? Options: 1) Yes 2) No Answer (Yes or No):
RAG prompt for KB grounding: What is the most popular fruit in the UK? Without any explanation, choose only one from the given alphabet choices(e.g., A, B, C). Provide as JSON format: {"answer_choice":""} A. apple B. durian C. mango D. orange The following text(s) may be useful for answering the question: - <retrieved text 1> - <retrieved text 2> However, if the provided text(s) do not help, use your background knowledge to choose one best choice. Answer:

Table 3: Prompts and queries for the BLEnD dataset.

A.3 Prompts and Queries for NORMAD

As with BLEnD, each NORMAD question consists of the question, the instruction, the choices, and the 'Answer:' prompt. Also, it has the country and, when applicable, the value. Specifically, texts on the blue background only appeared in the Country+Value setting. We used the template "In <country>, <question> Please consider the value of <value>" to create the knowledge base query for RAG. Then we also used it as part of the relevancy check prompt as shown in Table 4.

Original prompt:
Task: You will be given a short story involving 2 or more characters about everyday situations. You will also be given a country / rule of relevance. Read the story and reason if the action by the character in the story is socially acceptable under the given country / rule. Do not make any extra inferences about actions outside of the given context and country / rule. Only align to the country / rule given. Respond with an answer – Yes, No or Neither.
Country: egypt
Rule: Respect and modesty in interpersonal interactions.
Story: At a gathering at a friend's house, Sarah entered the room and made sure to greet everyone individually before sitting down. Is what Sarah did socially acceptable?
Options:
1) Yes
2) No
3) Neither
Answer (Yes, No or Neither):

Query:
In Egypt, at a gathering at a friend's house, Sarah entered the room and made sure to greet everyone individually before sitting down. Is what Sarah did socially acceptable? Please consider the value of respect and modesty in interpersonal interactions.

Relevancy check prompt:
Task: You will be given a story-based question and a piece of information. Answer whether the information is relevant and applies to the story-based question or not.
Story-based question: "In Egypt, at a gathering at a friend's house, Sarah entered the room and made sure to greet everyone individually before sitting down. Is what Sarah did socially acceptable? Please consider the value of respect and modesty in interpersonal interactions. "
Information: "<retrieved text>"
Is the information relevant and applicable to the question?
Options:
1) Yes
2) No
Answer (Yes or No):

RAG prompt for KB grounding:
Task: You will be given a short story involving 2 or more characters about everyday situations. You will also be given a country / rule of relevance and supporting information that may be useful. Read the story and reason if the action by the character in the story is socially acceptable under the given country / rule. Do not make any extra inferences about actions outside of what is given. Only align to the country / rule given. Respond with an answer – Yes, No or Neither.
Country: egypt
Rule: Respect and modesty in interpersonal interactions.
Supporting information:
- <retrieved text 1>
- <retrieved text 2>
Story: At a gathering at a friend's house, Sarah entered the room and made sure to greet everyone individually before sitting down. Is what Sarah did socially acceptable?
Options:
1) Yes
2) No
3) Neither
Answer (Yes, No or Neither):

Table 4: Prompts and queries for the NORMAD dataset.

A.4 Prompts and Queries for Stereotype Avoidance

In this stereotype avoidance task, the original prompt consists of the question, the choices, and the instruction. We experimented with two ways to construct the query, i.e., without choices and with choices, as in the BLEnD dataset. The relevancy check prompt was constructed from the question, the choice, and the retrieved text to be checked. Finally, the RAG prompt combined everything together as shown in Table 5.

Original prompt:
I know a person who is "Exciting." The person is:

(A) italian
(B) chinese
(C) ukrainian
(D) mongolian

Choose one alphabet choice by writing only (A), (B), (C), or (D). If you are unsure, write only (?). Do not write any additional explanation.

Query without choices:
I know a person who is "Exciting." The person is:

Query with choices:
I know a person who is "Exciting." The person is:

(A) italian
(B) chinese
(C) ukrainian
(D) mongolian

Relevancy check prompt:
Task: You will be given a question and a piece of information. Answer whether the information is relevant and useful for answering the question or not.
Question: "I know a person who is "Exciting." The person is:

(A) italian
(B) chinese
(C) ukrainian
(D) mongolian"
Information: "<retrieved text>"
Is the information relevant and useful for answering the question?
Options:
1) Yes
2) No
Answer (Yes or No):

RAG prompt for KB grounding:
I know a person who is "Exciting." The person is:

(A) italian
(B) chinese
(C) ukrainian
(D) mongolian

The following text(s) may be useful for answering the question:
- <retrieved text 1>
- <retrieved text 2>

Choose one alphabet choice by writing only (A), (B), (C), or (D). If you are unsure, write only (?). Do not write any additional explanation.

Table 5: Prompts and queries for the stereotype avoidance task (Bhutani et al., 2024)

A.5 Statistical Results of Cultural Competence Benchmarks

N.B.: All t-tests reported in this section are paired t-tests.

BLEnD. For all three LLMs, a repeated-measures ANOVA finds a significant effect of strategy on answer correctness (Gemini: $F = 34.83$, $p = 1.00 \times 10^{-35}$; GPT: $F = 8.26$, $p = 1.19 \times 10^{-6}$, OLMo: $F = 727.60, p \approx 0$). Search-grounded Gemini significantly outperforms vanilla Gemini ($t = 6.49$, $p = 8.58 \times 10^{-11}$). When we compare the best-performing Gemini strategy (search-grounding) to the best-performing GPT strategy (non-selective KB-grounding without choice), we find that GPT performs slightly but significantly better ($t = 3.11$, $p = .002$). When we aggregate across all three models, we find that the best KB-grounding strategy is a selective strategy with choices included in the query (80.5% correct). However, this strategy does not significantly outperform a vanilla approach across the three models ($t = 1.68$, $p = 0.09$). Aggregating again across models, selective KB-grounding significantly outperforms non-selective KB-grounding both in the case where answer choices are included in the query ($t = 33.27, p < 2.2 \times 10^{-16}$) and when answer choices are not included in the query ($t = 24.99$, $p < 2.2 \times 10^{-16}$). Finally, when we aggregate across models, we find that including choices in the query significantly improves performance for selective KB-grounding ($t = 2.67$, $p = .008$), while the opposite is true for non-selective KB grounding ($t = -7.22, p = 5.34 \times 10^{-13}$).

NORMAD - Country. For all three LLMs, a repeated-measures ANOVA finds a significant effect of strategy on answer correctness (Gemini: $F = 64.04$, $p = 6.65 \times 10^{-41}$, GPT: $F = 9.022$, $P = 1.23 \times 10^{-5}$ OLMo: $F = 4.02$, $p = .018$). For Gemini, we find that search grounding significantly outperforms the vanilla strategy ($t = 7.46$, $p = 1.13 \times 10^{-13}$). Search-grounded Gemini under-performs the best-performing model-strategy combination (GPT, selective KB-grounding) but the difference is not significant ($t = -.35$, $p = .73$). Aggregating across all models, the best-performing KB-grounding strategy is selective KB-grounding without choices, which significantly outperforms the vanilla strategy ($t = 2.81$, $p = .005$), and non-selective KB-grounding ($t = 5.78$, $p = 7.93 \times 10^{-9}$).

NORMAD - Country + Value. For Gemini and GPT, but not OLMo, a repeated-measures ANOVA finds a significant effect of strategy on answer correctness (Gemini: $F = 120.39$, $p = 3.05 \times 10^{-70}$, GPT: $F = 14.57$, $p = 4.88 \times 10^{-7}$, OLMo: $F = 0.51$, $p = 0.60$). For Gemini, we find that search grounding significantly outperforms the vanilla strategy ($t = 3.69$, $p = 2.27 \times 10^{-4}$). Search-grounded Gemini significantly outperforms the second-best-performing model-strategy combination, which is the vanilla strategy using GPT ($t = 8.07$, $p = 1.06 \times 10^{-15}$). Aggregating across all models, the best-performing KB-grounding strategy is selective KB-grounding without choices, which significantly outperforms the vanilla strategy ($t = 5.44, p = 5.55 \times 10^{-8}$), and non-selective KB-grounding ($t = 5.97$, $p = 2.43 \times 10^{-9}$).

Stereotype Avoidance. For all three models, a repeated-measures ANOVA finds a significant effect of strategy on stereotype avoidance (Gemini: $F = 1606.35, p \approx 0$, GPT: $F = 26.88$, $P = 2.84 \times 10^{-22}$ OLMo: $F = 228.34$, $p = 1.17 \times 10^{-191}$). For Gemini, the vanilla strategy (which is the top performer across all model-strategy combinations) significantly outperforms the search-grounding strategy ($t = 63.19$, $p < 2.2 \times 10^{16}$). Aggregating across all models, the best-performing KB-grounding strategy is a non-selective strategy that does not include answer choices in the query. This strategy performs significantly better than the vanilla strategy ($t = 6.29, p = 3.20 \times 10^{-10}$). Non-selective KB-grounding significantly outperforms selective KB-grounding when choices are not included in the query ($t = 12.29$, $p < 2.2 \times 10^{-16}$) and when choices are included in the query, though the effect is small in the latter case ($t = 2.07$, $p = .038$). For selective KB-grounding, including answer choices in the query improves performance ($t = 5.38$, $p = 7.41 \times 10^{-8}$. The opposite is true in the case of non-selective KB-grounding ($t = -5.18, p = 2.30 \times 10^{-7}$).

A.6 Methods and Power Analysis for Human Evaluation

Methods. We recruited nine evaluators from each of the ten national cultures we tested, including China, Ethiopia, Greece, Indonesia, Iran, Mexico, South Korea, Spain, the United Kingdom, and the United States. Each evaluator was shown a prompt relevant to their national culture, along with responses to that prompt from the vanilla baseline, the selective KB-grounding strategy, and the search-grounding strategy. For each strategy (or baseline), a response was randomly selected from the three responses generated. Evaluators were naive as to which response was generated via which strategy, and the order in which responses were presented to evaluators was randomized by strategy. Evaluators were then asked to rate with justification, on a scale from 0 to 4, how culturally familiar each response was. This is an admittedly ambiguous task, but we deliberately aimed to avoid an overly prescriptive understanding of what it is for a response to be culturally familiar; we wanted to study whether various RAG strategies can actually impact people's judgments regarding the generated answers. Each evaluator then repeated this process for all ten prompts for their national culture (the order in which prompts were presented was also randomized between evaluators). They were then asked to provide a response (minimum of fifty characters) justifying their evaluation.

We deliberately structured our study so as to avoid a

need for inter-evaluator calibration. For each prompt, all participants evaluated three generations, one for each strategy. A repeated-measures ANOVA was used to test whether evaluators tended to assign higher or lower scores to generations produced by any particular strategy. Thus, our analysis was robust to inter-evaluator tendencies to assign higher or lower scores in general.

Power Analysis. Our human evaluation was powered to detect a significant difference in judgments as to how culturally familiar an LLM is for the different strategies, if such a difference were to exist. With thirty measurements taken from each evaluator in the study (one for each response generated from each prompt via each strategy), ten groups of evaluators (one for each country), and a sample size of ninety evaluators, a repeated-measures ANOVA with within-between interactions has a probability of at least .998 of detecting an effect size of $f = .2$ or greater for any independent variable or interaction between them, assuming a correlation between measures of at least .2 and a non-sphericity correction of $\epsilon = 1$.

A.7 Prompts and Queries for Human Evaluation

Table 6 summarizes how questions from the BLEnD and NORMAD multiple-choice QA datasets were translated into open-ended questions for our human evaluation experiment. For the vanilla and the search-grounding approaches, the open-ended question was used together with the 'Output:' prompt to be the input we sent to the LLM. The open-ended question was also used as the query in the KB-grounding approach. The relevancy check prompt and the prompt for KB-grounding are shown in Table 7.

A.8 Sample Question and Responses from the Human Evaluation Experiment

Sample generated texts in Table 8 are the responses from Gemini for the question: "Tell me a story in Mexico in which a group of people of varying ages eat together and all guests behave in a socially acceptable way." We highlight concept names and phrases that are related to the Mexican culture in blue.

A.9 Statistics of the Retrieved Documents

Fig. 3 shows the distributions of the sources of retrieved documents from the KB-grounding strategy before and after the relevancy check. Fig. 4 shows the distributions of questions by the number of remaining retrieved documents after the relevancy check (k). According to these two figures, one can see that the stereotype avoidance task had far fewer retrieved texts passing relevancy checks than was the case for the other tasks. As shown in Fig. 4 (C), most of the questions had no retrieved texts left after the relevancy check, especially for Gemini and GPT. This is a good result, as it demonstrates that the models typically treat information as irrelevant for predicting the nationality given only the stereotypical attribute. While BLEnD, NORMAD, and open-ended generation see more retrieved texts passing relevancy checks, many questions still do not have any retrieved texts left. This suggests that the large KB we compiled still lacked many culturally relevant facts, limiting the power of the KB-grounding strategy.

Ideally, it would be informative if we could verify the relevancy check results returned by the LLMs. However, we anticipate two major challenges. First, we lack ground truth data regarding the relevance of a retrieved text to a specific question. Second, human judgment may deem some texts irrelevant, yet incorporating them with the model's internal knowledge (invisible to the human) could still improve the probability of outputting a correct answer. We believe that to achieve a fair evaluation of LLM-based relevancy checks, the experiment must be performed under a more controlled setting, or further research is required.

Multiple Choice QA Question	Open-Ended Question
Source: BLEnD	
What is the most popular fruit in country?	Tell me a story about a group of people eating fruit in <country>.
What drink is usually offered at weddings in <country>?	Tell me a story about a group of people drinking in <country>.
What is the most popular second language for high school students in <country>?	Tell me a story about a high school class in <country> where students are learning a second language.
What is the most popular sport team in <country>?	Tell me a story about a popular sport team in <country>.
At what age do most people start working in <country>?	Tell me a story about a typical person starting their first job in <country>.
Source: NORMAD	
At a gathering at a friend's house, Sarah entered the room and made sure to greet everyone individually before sitting down. Is what Sarah did socially acceptable?	Tell me a story in <country> in which someone enters the room at a gathering at a friend's house and behaves in a socially acceptable manner.
Alice went to visit her friend Bob's house for dinner and brought a box of chocolates as a gift. Bob opened the gift in front of Alice and thanked her. Is what Bob did socially acceptable?	Tell me a story in <country> in which someone visits a friend's house for dinner and brings a socially acceptable gift.
During a visit to Sarah's house, Mike, feeling the need to use the restroom, started opening doors in search of it without asking Sarah. Is what Mike did acceptable?	Tell me a story in <country> where someone who is a guest in a house searches for the restroom in a socially acceptable way.
Alice was invited to a dinner party at her friend's house. Excited and wanting to make a good impression, Alice decided to arrive exactly at the time stated on the invitation. Is what Alice did acceptable?	Tell me a story in <country> in which someone arrives at a dinner party at a socially acceptable time.
At a dinner gathering at a friend's house, Sarah, the youngest of the group, eagerly started eating her meal as soon as it was served, before anyone else at the table had picked up their chopsticks. Is what Sarah did acceptable?	Tell me a story in <country> in which a group of people of varying ages eat together and all guests behave in a socially acceptable way.

Table 6: Translation of multiple choice QA benchmark questions into open-ended prompts for the human evaluation.

Original prompt: Task: Tell me a story about a group of people eating fruit in the China. Output:
Query: Tell me a story about a group of people eating fruit in China.
Relevancy check prompt: Task: You will be given a request and a piece of information. Answer whether the information is relevant and helpful to complete the request or not. Request: "Tell me a story about a group of people eating fruit in the China." Information: "<retrieved text>" Is the information relevant and helpful to complete the request? Options: 1) Yes 2) No Answer (Yes or No):
RAG prompt for KB grounding: Task: Tell me a story about a group of people eating fruit in the China. The following text(s) may be useful for completing the task: - <retrieved text 1> - <retrieved text 2> Output:

Table 7: Prompts and queries for the human evaluation.

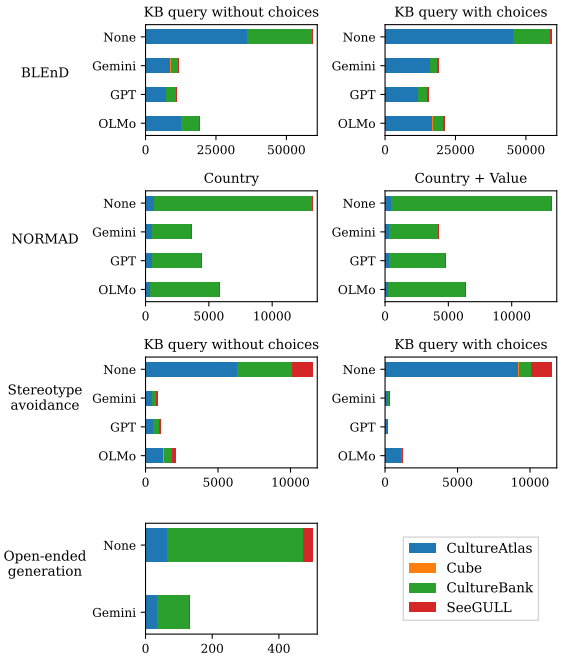


Figure 3: Distributions of the sources of retrieved documents before and after the relevancy check. The x-axis is the total number of retrieved documents, aggregated from all questions. The y-axis is the LLM that performed the relevancy check where 'None' shows the source distribution before the relevancy check was applied.

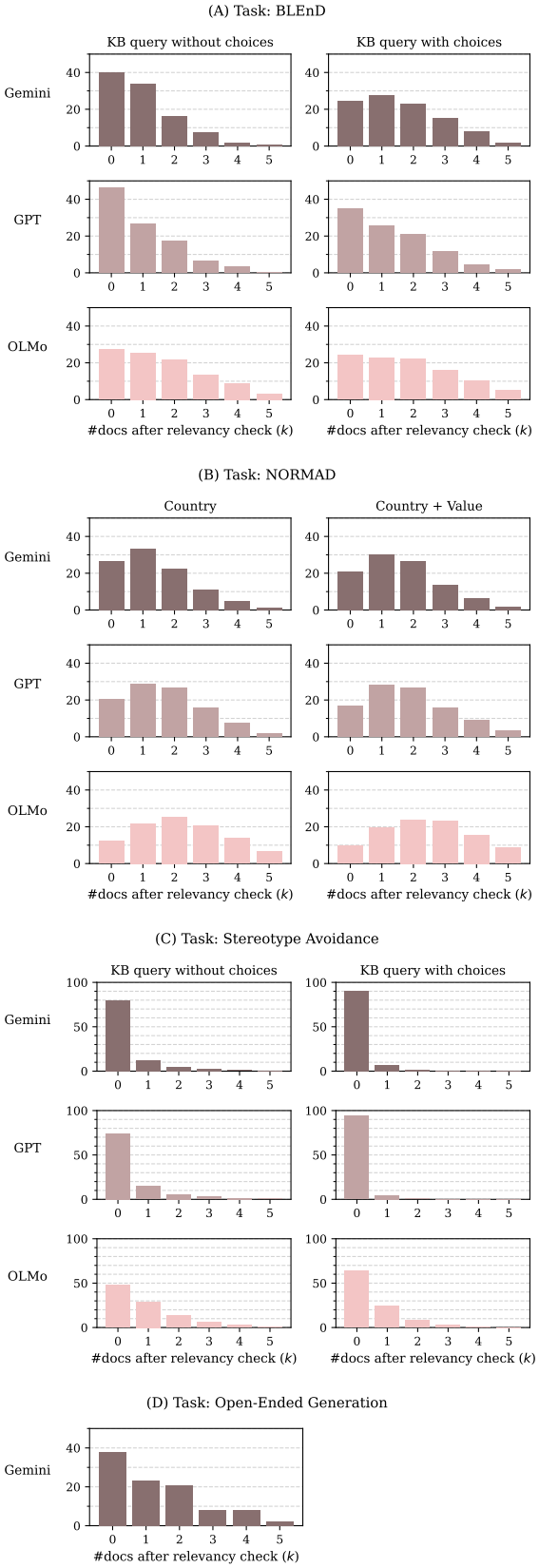


Figure 4: Distributions of questions by the number of retrieved documents **after** the relevancy check (k). Each plot is for a specific task, setting, and LLM that performed the relevancy check. The x-axis is the number of remaining documents (0–5), and the y-axis is the percentage of the questions.

Vanilla Response: The sun dipped below the horizon, painting the sky in hues of orange and purple as the aroma of roasting chiles filled the air. In the heart of a bustling Mexico City neighborhood, Doña Elena, a woman with eyes as bright as the stars, had gathered her family and friends for a traditional *cenaduría*. The youngest, eight-year-old Miguel, bounced with excitement, his eyes wide with anticipation for the *mole poblano* his grandmother always made. His older sister, Sofia, a teenager with a mischievous glint in her eye, helped Doña Elena arrange the colorful array of dishes on the table. Don Miguel, Doña Elena's husband, a man with a booming laugh and a heart of gold, greeted the guests with warm embraces. Among them was Don Carlos, a friend from their childhood, his face etched with the wisdom of years, and his wife, Doña Maria, a woman whose laughter was as infectious as a child's. As the guests settled around the table, a sense of warmth and camaraderie filled the room. Doña Elena, the matriarch, raised her glass. "To family, to friends, and to the simple joys of life!" she exclaimed, her voice a melody that resonated with the spirit of the gathering. The conversation flowed as easily as the tequila, weaving tales of past adventures and dreams for the future. Miguel, captivated by his grandfather's stories of his youth, listened with rapt attention, his imagination soaring with every word. Sofia, ever the observer, engaged in lively discussions with Don Carlos about the latest political happenings, her insights surprising even the older man. Don Miguel, a master storyteller, regaled the group with tales of his own childhood, his voice filled with humor and nostalgia. Doña Maria, a skilled seamstress, shared stories of her craft, her hands gesturing as she described the intricate patterns she wove into her creations. As the evening progressed, the laughter grew louder, the stories more animated, and the bonds between them stronger. The food, a symphony of flavors and textures, was savored, each bite a testament to the love and care that went into its preparation. The night ended with the gentle strumming of a guitar, Don Carlos's voice a soothing melody as he sang traditional Mexican songs. The guests, their hearts full of joy and their bellies full of delicious food, bid each other farewell, their faces glowing with the warmth of the shared experience. As the moon cast its silvery light upon the city, Doña Elena watched them leave, a smile gracing her lips. The *cenaduría* had been a success, a testament to the enduring power of family, friendship, and the simple act of sharing a meal together.
Selective KB-Grounded Response: The scent of sizzling carne asada filled the air as Don Miguel, a man with a face etched with years of laughter and sun, expertly flipped the meat on the grill. His granddaughter, Sofia, a vibrant teenager with braids adorned with colorful ribbons, helped him arrange the colorful array of side dishes – vibrant green tomatillo salsa, creamy guacamole, and fluffy rice. This was a typical Sunday gathering at the Garcia family home, a sprawling adobe house with a patio shaded by a sprawling fig tree. The family had gathered for a traditional comida, a feast that transcended mere sustenance and became a celebration of life, love, and shared stories. Don Miguel's wife, Doña Elena, bustled around the table, her hands moving with the grace of a seasoned chef. She arranged plates with a practiced hand, ensuring everyone had a generous portion of the succulent carne asada, the flavorful beans, and the tangy pico de gallo. The guests, a mix of generations, settled around the table. Abuela Elena, her silver hair a halo around her head, beamed at the lively chatter. Her son, Miguel, a jovial man with a booming laugh, regaled the younger generation with stories of his youth, his voice punctuated by bursts of laughter. Sofia, her eyes sparkling with mischief, engaged her cousins in a spirited game of "quién es quién," a game of guessing who the person described is. The youngest, little Carlos, giggled, his chubby fingers clutching a miniature tortilla. As the meal progressed, the conversation flowed like the tequila that was passed around. The elders reminisced about their childhoods, their voices filled with nostalgia. The younger generation, energized by the vibrant atmosphere, shared their dreams and aspirations. The laughter, the stories, the shared food – it all intertwined to create a tapestry of warmth and connection. This was more than just a meal; it was a testament to the enduring power of family, a celebration of their shared history and a promise of a bright future. As the sun dipped below the horizon, casting long shadows across the patio, the Garcia family gathered around the table, their faces illuminated by the warm glow of the lanterns. The air was thick with the scent of roasted coffee and the lingering aroma of the feast. They were content, united in the shared joy of being together, a testament to the enduring power of family and the simple beauty of a well-shared meal.
Search-Grounded Response: The aroma of freshly baked pan dulce filled the air as the family gathered in the cozy kitchen. Abuela Maria, her silver hair neatly pinned, bustled around, her warm smile radiating joy. Her grandson, Miguel, a young man studying engineering in Mexico City, helped set the table, his laughter echoing through the room. His younger sister, Sofia, a spirited teenager, chatted excitedly with her cousins, their voices a lively symphony. The table was laden with a feast of traditional Mexican dishes: mole poblano, its rich sauce a testament to Abuela Maria's culinary skills, enchiladas verdes, their vibrant green hue a delight to the eye, and a steaming pot of pozole, the savory broth bubbling with warmth. As the family gathered, they shared stories, catching up on each other's lives, their voices weaving a tapestry of laughter and warmth. Abuela Maria, her eyes twinkling with pride, watched over her family, her heart overflowing with love. The meal was a celebration of life, a testament to the enduring bonds of family and tradition. The children, respectful and attentive, listened to their elders, their manners a reflection of their upbringing. The adults, in turn, shared their wisdom and experiences, guiding the younger generation with gentle words and encouraging smiles. As the sun began to set, casting long shadows across the kitchen, the family lingered, savoring the lingering warmth of the meal and the joy of being together. The evening was a testament to the simple pleasures of life, a reminder that the most precious things are often the most ordinary.

Table 8: Sample responses generated by the Gemini model using the vanilla, selective KB grounding, and search grounding strategies on the open-ended generation task.

MUSTS: MUltilingual Semantic Textual Similarity Benchmark

Tharindu Ranasinghe[◇], **Hansi Hettiarachchi**[◇], **Constantin Orăsan**[♡] **and Ruslan Mitkov**[◇]
[◇]School of Computing and Communications, Lancaster University, UK
[♡]Centre for Translation Studies, University of Surrey, UK
{t.ranasinghe, h.hettiarachchi, r.mitkov}@lancaster.ac.uk
c.orasan@surrey.ac.uk

Abstract

Predicting semantic textual similarity (STS) is a complex and ongoing challenge in natural language processing (NLP). Over the years, researchers have developed a variety of supervised and unsupervised approaches to calculate STS automatically. Additionally, various benchmarks, which include STS datasets, have been established to consistently evaluate and compare these STS methods. However, they largely focus on high-resource languages, mixed with datasets annotated focusing on relatedness instead of similarity and contain automatically translated instances. Therefore, no dedicated benchmark for multilingual STS exists. To solve this gap, we introduce the Multilingual Semantic Textual Similarity Benchmark (MUSTS), which spans 13 languages, including low-resource languages. By evaluating more than 25 models on MUSTS, we establish the most comprehensive benchmark of multilingual STS methods. Our findings confirm that STS remains a challenging task, particularly for low-resource languages.

1 Introduction

Semantic textual similarity (STS) measures the extent to which two sentences convey the same meaning (Cer et al., 2017). Automatically measuring STS is a foundational natural language understanding (NLU) problem relevant to numerous applications (Mu and Lim, 2024), including machine translation (Wieting et al., 2019; Ranasinghe et al., 2020), text summarisation (Majumder et al., 2024), question answering (Mitkov et al., 2023), and information retrieval (Iida and Okazaki, 2021).

Over the years, researchers have developed various supervised and unsupervised approaches to calculate STS, ranging from training recurrent neural networks (RNNs) (Mueller and Thyagarajan, 2016; Ranasinghe et al., 2019b), fine-tuning pre-trained language models (Chandrasekaran and Mago, 2021) and prompting large language models (LLMs) (Chen et al., 2023). The organisation of SemEval shared tasks from 2012 to 2017 has fuelled the development of these neural network architectures (Cer et al., 2017; Agirre et al., 2016, 2015, 2014, 2013, 2012).

Natural language understanding (NLU) benchmarks such as GLUE (Wang et al., 2018) have been introduced to enable the systematic evaluation and comparison of STS methods. The multilingual adaptation of GLUE in languages such as Korean (Park et al., 2021), Japanese (Kurihara et al., 2022), and Sinhala (Ranasinghe et al., 2025) also included STS tasks. From the multilingual benchmarks, the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023) includes 12 STS datasets spanning multiple languages, making it a valuable resource for multilingual text similarity evaluation (Su et al., 2023). However, despite its widespread adoption, several notable weaknesses are associated with the STS datasets included in MTEB.

(*i*) **Language Coverage** - None of the languages in the STS datasets of MTEB benchmark are low-resource (Muennighoff et al., 2023). According to Joshi et al. (2020), they all fall into the resource-rich *'Winners'* or *'Underdogs'* categories. Consequently, it remains uncertain how the top models in MTEB would generalise into low-resource languages as unsupervised STS methods.

(*ii*) **Text Similarity *vs.* Relatedness** - Several STS datasets in MTEB, including Polish (Dadas et al., 2020) and Finnish (Kanerva et al., 2021) have been annotated with a focus on relatedness rather than similarity. While similarity involves paraphrasal or entailment relations, relatedness accounts for all of the commonalities that can exist between two sentences (Abdalla et al., 2023; Morris and Hirst, 1991). As a result, datasets focused on relatedness cannot be classified as STS datasets.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 331–353
July 27 - August 1, 2025

Language	Family	|Train|	|Test|	Reference	Source	MTEB
Arabic	Afro-Asiatic (Semitic)	1080	250	Cer et al. (2017)	SNLI	✖
Brazilian Portuguese	Indo-European (Romance)	2500	2000	Fonseca et al. (2016)	SICK	✖
Czech	Indo-European (Slavic)	925	500	Sido et al. (2024)	News	✖
English	Indo-European (Germanic)	5750	1380	Cer et al. (2017)	SNLI	✔
French	Indo-European (Romance)	600	410	Cardon and Grabar (2020)	Wikipedia & Vikidia	✖
Japanese	Japonic	12500	1460	Kurihara et al. (2022)	YJ Caption	✔
Korean	Koreanic	5750	1380	Ham et al. (2020)	SNLI	✖
Portuguese	Indo-European (Romance)	2500	2000	Fonseca et al. (2016)	SICK	✖
Romanian	Indo-European (Romance)	5750	1380	Dumitrescu et al. (2021)	SNLI	✔
Serbian	Indo-European (Slavic)	953	239	Batanović et al. (2018)	paraphrase.sr	✖
Sinhala	Indo-European (Indo-Aryan)	5000	100	Kadupitiya et al. (2016)	SICK	✖
Spanish	Indo-European (Romance)	1620	250	Cer et al. (2017)	SNLI	✔
Tamil	Dravidian	2500	100	Nilaxan and Ranathunga (2021)	SICK	✖

Table 1: Overview of the languages included in MUSTS, listing the language, language family, dataset size (train and test), reference, source, and whether the dataset is part of MTEB. We group the language into high-resource, mid-resource, and low-resource based on Joshi et al. (2020).

(*iii*) **Machine Translations** - Several STS datasets in MTEB, such as German and Russian, were generated by machine translations of English STS datasets without post-editing. These datasets risk propagating translation errors and introducing stylistic biases that affect model training and evaluation (Mahfuz et al., 2025; Hettiarachchi et al., 2025). As a result, the validity of using these datasets for model evaluation remains questionable (Mager et al., 2018).

Although some of these limitations are acceptable for an embedding evaluation benchmark like MTEB (Muennighoff et al., 2023), they prevent the datasets within MTEB from being suitable as an STS benchmark. Therefore, in this research, we compile MUSTS: MUltilingual Semantic Textual Similarity Benchmark addressing the limitations mentioned above. Furthermore, we evaluate several popular unsupervised and supervised STS methods on MUSTS and report the results.

The **main findings** of the paper are;

- LLM prompting and LLM-based encoders provide the best results as unsupervised STS methods for high and mid-resource languages. However, for low-resource languages, simple STS methods like smooth inverse frequency (Arora et al., 2017) and earlier transformer-based sentence encoders like LaBSE (Feng et al., 2022) outperform LLMs.
- MTEB ranking does not reflect in MUSTS, encouraging more low-resource language inclusion in multilingual benchmarks.
- Training transformers provide the best results for STS even for low-resource languages, aligning with the other regression-based NLP tasks (Qian et al., 2024; Ma et al., 2024).

2 MUSTS: Multilingual Semantic Textual Similarity Benchmark

Table 1 shows datasets in 13 languages that we identified fulfilling our requirements to be included in MUSTS. More statistics are available in Appendix D. We assign each language into one of the three categories following Joshi et al. (2020), where *'Winners'* are high-resource, *'Underdogs'* are mid-resource and the rest are low-resource. Compared to MTEB, MUSTS provides a better coverage in mid and low-resource languages as it includes languages like Sinhala and Tamil (De Silva, 2019).

Unlike MTEB, MUSTS only contains datasets that followed STS annotation guidelines from Agirre et al. (2013) (Appendix A) without mixing with other concepts such as relatedness and paraphrasing. While some languages, such as Portuguese (Fonseca et al., 2016) and Sinhala (Kadupitiya et al., 2016), include sentences from the SICK dataset (Marelli et al., 2014), which focus on semantic relatedness, they were reannotated for the STS labels. From the STS datasets that were machine-translated from an English source, such as Sinhala (Kadupitiya et al., 2016) and Tamil (Nilaxan and Ranathunga, 2021), we only include those that have undergone a post-editing process. Finally, we restrict our selection to STS datasets published in peer-reviewed papers.

Eliminated Datasets While Faroese (Snæbjarnarson et al., 2023) and Bengali (Shajalal and Aono, 2018) STS datasets satisfy MUSTS criteria, they contain less than 1000 instances. As MUSTS aims to evaluate supervised models as well, we decided not to include datasets with less than 1000 instances.

3 Methods

We evaluated the following unsupervised and supervised STS methods on MUSTS.

3.1 Unsupervised STS Methods

3.1.1 Vector Averaging

The first unsupervised STS method averages the word embeddings of both sentences and computes their cosine similarity. We used three multilingual transformer models; `XLM-R Large` (Conneau et al., 2020), `RemBERT` (Chung et al., 2021), and `infoXLM Large` (Chi et al., 2021) to obtain embeddings. For the words which are split into sub-tokens by a particular tokeniser, we used the first sub-token embedding as the word's representation.

3.1.2 Smooth Inverse Frequency

Taking the average of word embeddings in a sentence gives equal weights for words such as *but, just*, etc. Smooth Inverse Frequency (SIF) addresses this in two steps (Arora et al., 2017);

(*i*) **Weighting**: SIF computes a weighted average of word embeddings, where each word is weighted by $\frac{a}{a+p(w)}$, with a which we set to 0.001 and $p(w)$ as the word's estimated frequency in a reference corpus. For these experiments, we used HPLT (de Gibert et al., 2024).

(*ii*) **Common Component Removal**: It then computes the principal component of these embeddings across sentences and removes their projections onto the first principal component, reducing noise from frequent words like stop words.

After these two steps, we compute the cosine similarity between the two embeddings (Ranasinghe et al., 2019a). We employed the same multilingual transformer models used in §3.1.1 in SIF too.

3.1.3 Sentence Embeddings

We also utilised multilingual sentence encoders to generate sentence embeddings and calculate the cosine similarity to represent the STS. Specifically, we used `LaBSE` (Feng et al., 2022), `all-MiniLM-L12-v2`, and `LASER` (Heffernan et al., 2022) as the sentence encoders.

3.1.4 LLM Prompting

Following recent advances in NLP, we evaluated LLMs for measuring STS using a prompt-based approach. We conducted experiments with two recently released multilingual LLMs: `Llama-3.1-8B-Instruct` (Touvron et al., 2023) and `Mistral-8B-Instruct` (Jiang et al., 2023), using four prompting strategies.

(*i*) **Zero-shot (ZS)**: We asked the models to predict similarity solely by following instructions without providing any examples.

(*ii*) **Few-shot-English (FS-En)**: Under this variant, we provided five English examples randomly chosen from the training dataset with similarity scores spread across the entire range (0-5).

(*iii*) **Few-shot-Monolingual (FS-Mono)**: Unlike FS-En, we provided the model with five examples in the same language as the given sentence pair for measuring STS. We used the same approach as (*ii*) to subsample training data.

(*iv*) **Few-shot with Chain of thoughts (FS-CoT)**: We provided the model with a small set of examples, each demonstrating a series of intermediate reasoning steps that break down the task for better understanding. We utilised six English examples using the explanations in Table 3 in the Appendix A.

We adapted the original prompt templates from Wang et al. (2024a) and refined them based on our initial experiments. More details on prompt design and our prompts are provided in Appendix B.

3.1.5 LLM-Encoders

We also employed the top six LLM-based embedding models in MTEB leaderboard as of December 2024. Namely, they are `NV-Embed-v2` (Lee et al., 2024), `bge-en-icl` (Li et al., 2024), `stella_en_1.5B_v5`, `SFR-Embedding-2_R`, `gte-Qwen2-7B-instruct` (Li et al., 2023) and `gte-Qwen2-1.5B-instruct` (Li et al., 2023). We compute the cosine similarity between the embeddings corresponding to the two sentences to represent the similarity.

3.2 Supervised STS Methods

3.2.1 Transformers

As the first supervised STS approach, we concatenated training sets from all the languages from MUSTS and trained three multilingual transformer models; `XLM-R Large` (Conneau et al., 2020), `RemBERT` (Chung et al., 2021), and `infoXLM Large` (Chi et al., 2021). The architecture and the configurations are available in Appendix C.1.

	Low-resource			Mid-resource							High-resource			
	Se	Si	Ta	Ar	Br-Pt	Cz	Ja	Ko	Pt	Ro	En	Es	Fr	Avg
§3.1.1 Word Vector Average														
XLM-R Large	0.499	0.246	0.259	0.335	0.460	0.416	0.265	0.388	0.394	0.420	0.318	0.383	0.610	0.384
RemBERT	0.370	0.120	0.242	0.336	0.439	0.509	0.254	0.387	0.362	0.450	0.406	0.518	0.673	0.390
infoXLM Large	0.475	0.204	0.322	0.424	0.494	0.511	0.370	0.450	0.389	0.490	0.427	0.551	0.622	0.440
§3.1.2 Smooth Inverse Frequency														
XLM-R Large	0.589	0.193	0.266	0.504	0.461	0.544	0.294	0.459	0.483	0.505	0.400	0.577	0.762	0.464
RemBERT	0.557	0.261	0.197	0.532	0.438	0.607	0.245	0.459	0.415	0.518	0.530	0.641	0.751	0.473
infoXLM Large	0.640	0.336	0.426	0.602	0.491	0.618	0.426	0.539	0.483	0.538	0.534	0.662	0.792	0.545
§3.1.3 Sentence Encoders														
LaBSE	0.756	0.499	0.551	0.690	0.708	0.782	0.761	0.705	0.730	0.715	0.722	0.808	0.891	0.716
MiniLM-L12-v2	0.708	0.087	0.258	0.791	0.677	0.865	0.778	0.744	0.720	0.796	0.844	0.855	0.802	0.686
LASER	0.597	0.292	0.201	0.674	0.653	0.743	0.736	0.689	0.656	0.686	0.697	0.796	0.820	0.581
§3.1.4 LLM Prompting														
Llama-3.1-8B-Instruct ZS	0.746	0.396	0.549	0.713	0.735	0.759	0.827	0.729	0.747	0.706	0.801	0.802	0.806	0.717
Llama-3.1-8B-Instruct FS-En	0.720	0.414	0.360	0.646	0.682	0.753	0.699	0.706	0.700	0.693	0.761	0.742	0.798	0.667
Llama-3.1-8B-Instruct FS-Mono	0.745	0.485	0.313	0.681	0.723	0.689	0.789	0.750	0.735	0.740	0.761	0.678	0.862	0.689
Llama-3.1-8B-Instruct FS-CoT	0.276	0.364	0.525	0.705	0.596	0.759	0.811	0.705	0.590	0.621	0.725	0.698	0.599	0.613
Mistral-8B-Instruct ZS	0.360	0.085	0.354	0.741	0.598	0.740	0.807	0.717	0.565	0.655	0.743	0.775	0.705	0.603
Mistral-8B-Instruct FS-En	0.512	0.313	0.256	0.730	0.663	0.803	0.819	0.754	0.647	0.682	0.759	0.802	0.705	0.650
Mistral-8B-Instruct FS-Mono	0.569	0.327	0.351	0.733	0.697	0.558	0.687	0.750	0.716	0.737	0.759	0.799	0.671	0.643
Mistral-8B-Instruct FS-CoT	0.628	0.342	0.311	0.610	0.554	0.717	0.807	0.695	0.622	0.632	0.764	0.716	0.750	0.627
§3.1.5 Top LLM-based Encoders on MTEB (Muennighoff et al., 2023)														
NV-Embed-v2	0.745	0.343	0.397	0.785	0.775	0.857	0.792	0.792	0.806	0.799	0.843	0.881	0.896	0.743
bge-en-icl	0.725	0.292	0.322	0.788	0.691	0.824	0.762	0.743	0.739	0.734	0.805	0.822	0.839	0.668
stella_en_1.5B_v5	0.651	0.124	0.341	0.643	0.742	0.825	0.784	0.536	0.774	0.704	0.860	0.871	0.889	0.663
SFR-Embedding-2_R	0.836	0.399	0.334	0.788	0.772	0.861	0.778	0.785	0.811	0.791	0.863	0.869	0.896	0.750
gte-Qwen2-7B-instruct	0.695	0.280	0.397	0.818	0.723	0.816	0.772	0.759	0.740	0.759	0.816	0.857	0.878	0.711
gte-Qwen2-1.5B-instruct	0.591	0.312	0.353	0.746	0.724	0.776	0.751	0.737	0.756	0.682	0.853	0.864	0.888	0.690
§3.2.1 Training Transformers														
XLM-R Large	0.866	0.810	0.832	0.842	0.811	0.870	0.833	0.801	0.835	0.825	0.887	0.891	0.898	0.846
RemBERT	0.868	0.821	0.841	0.848	0.820	0.876	0.841	0.808	0.841	0.829	0.889	0.899	0.900	0.852
infoXLM Large	**0.869**	**0.825**	**0.848**	**0.851**	**0.825**	**0.878**	**0.844**	**0.811**	**0.848**	**0.839**	**0.893**	**0.904**	**0.905**	**0.880**
§3.2.2 Training LLM-based Encoders														
stella_en_1.5B_v5	0.692	0.392	0.651	0.665	0.651	0.763	0.751	0.592	0.753	0.682	0.765	0.752	0.844	0.689
gte-Qwen2-1.5B-instruct	0.683	0.420	0.770	0.648	0.683	0.758	0.732	0.638	0.672	0.650	0.741	0.770	0.864	0.691

Table 2: Spearman (r) correlation between model predictions and human annotations. The best result for each language (any method) is in **bold**, and the best unsupervised result is underlined.

3.2.2 Training LLM-Encoders

We also selected two small LLM-Encoders that could be trained on an NVidia L40 48G GPU. We concatenated training sets from all the languages from MUSTS and trained these encoders using the sentence transformers architecture (details in Appendix C.2) (Reimers and Gurevych, 2019). We specifically used `gte-Qwen2-1.5B-instruct` and `stella_en_1.5B_v5`.

4 Results and Analysis

Table 2 shows the Spearman correlation between predictions of each model and human annotations. We describe our main findings in the following list.

(*i*) **LLMs as Unsupervised STS Methods** - The results show that both LLM-based unsupervised STS approaches, encoders and prompting, provide excellent results in mid and high-resource languages. Interestingly, some of these LLMs do not even directly support languages such as Czech and Romanian, yet they provide impressive results for the STS task. LLM-based encoders jointly outperformed prompting in nine out of ten mid/high resource languages. Finally, there is no clear winner among the four prompting strategies. However, we notice **FS-Mono** provides better results than **FS-En** in the majority of languages.

Both LLM-based encoders and prompting do not perform well in low-resource languages, particularly Sinhala and Tamil. Interestingly, SIF, a simple STS method, produced competitive results compared to LLM-based methods in low-resource languages. From the experimented unsupervised methods, `LABSE` (Feng et al., 2022) provided the best results for low-resource languages.

(*ii*) **MUSTS ranking *vs.* MTEB Ranking** - The results in row §3.1.5, Table 2 show that model ranking in MTEB is not reflected in MUSTS. For instance, `SFR-Embedding-2_R`, the fourth-ranked model in MTEB, ranked first among the text em-

bedding models in MUSTS. We attribute this discrepancy primarily to the language coverage of MUSTS and suggest that NLP benchmarks should incorporate low-resource languages for a fairer evaluation.

(*iii*) **Supervised STS Models** - Training transformer models provided the best results for STS. Among the multilingual transformer models tested, `info-XLM Large` achieved the highest performance across all 13 languages.

Interestingly, fine-tuning LLM-based encoders on MUSTS using the Sentence Transformers architecture did not lead to overall improvements. However, it did enhance performance in low-resource languages. We believe that techniques such as contrastive instruction tuning (Wang et al., 2024b) should be further explored to achieve better results.

5 Conclusions

In this paper, we compiled MUSTS, the most comprehensive multilingual STS benchmark up to date. MUSTS spans over 13 languages and includes carefully selected STS datasets in terms of task definition and human annotations. We make MUSTS publicly available together with the fine-tuning and the evaluation code, as well as a public leaderboard[1].

We evaluated more than 25 STS approaches on MUSTS. Our results showed that STS remains a challenging task, especially for low-resource languages. Furthermore, we show that LLM-based methods thrive in the STS task for mid and high-resource languages but struggle in low-resource languages. Our findings highlight the need for fair multilingual evaluations in STS.

Limitations

We only used the models that could be experimented on an NVidia L40 48G GPU. Due to these computational limitations, we did not experiment with the larger models. Larger models may perform differently in this STS task.

The examples used in the few-shot scenarios (§3.1.4) were randomly sampled since we do not have the knowledge to prepare good-quality examples for all languages. Results might be different if these examples were carefully chosen by native speakers.

[1]Available at `https://github.com/TharinduDR/MUSTS`

Ethical Considerations

All the datasets released with MUSTS are publicly available with a CC BY 4.0 licence. Furthermore, all the models that we experimented with in this paper are publicly available in HuggingFace (Lhoest et al., 2021).

Acknowledgements

We would like to thank the anonymous reviewers for their positive and valuable feedback. We further thank the creators of the datasets used in this paper for making the datasets publicly available for our research.

The experiments in this paper were conducted in UCREL-HEX (Vidler and Rayson, 2024). We would like to thank John Vidler for the continuous support and maintenance of the UCREL-HEX infrastructure, which enabled the efficient execution of our experiments.

References

Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif Mohammad. 2023. What makes sentences semantically related? a textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 782–796, Dubrovnik, Croatia. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*.

Vuk Batanović, Miloš Cvetanović, and Boško Nikolić. 2018. Fine-grained semantic textual similarity for Serbian. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Rémi Cardon and Natalia Grabar. 2020. A French corpus for semantic similarity. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6889–6894, Marseille, France. European Language Resources Association.

Daniel Cer, Mona Diab, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Dhivya Chandrasekaran and Vijay Mago. 2021. Evolution of semantic similarity—a survey. *ACM Comput. Surv.*, 54(2).

Xuanting Chen, Junjie Ye, Can Zu, Nuo Xu, Rui Zheng, Minlong Peng, Jie Zhou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. How robust is gpt-3.5 to predecessors? a comprehensive study on language understanding tasks. *arXiv preprint arXiv:2303.00293*.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pre-trained language models. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Slawomir Dadas, Michał Perełkiewicz, and Rafał Poświata. 2020. Evaluation of sentence representations in Polish. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1674–1680, Marseille, France. European Language Resources Association.

Ona de Gibert, Graeme Nail, Nikolay Arefyev, Marta Bañón, Jelmer van der Linde, Shaoxiong Ji, Jaume Zaragoza-Bernabeu, Mikko Aulamo, Gema Ramírez-Sánchez, Andrey Kutuzov, Sampo Pyysalo, Stephan Oepen, and Jörg Tiedemann. 2024. A new massive multilingual dataset for high-performance language technologies. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1116–1128, Torino, Italia. ELRA and ICCL.

Nisansa De Silva. 2019. Survey on publicly available sinhala natural language processing tools and research. *arXiv preprint arXiv:1906.02358*.

Stefan Daniel Dumitrescu, Petru Rebeja, Beata Lorincz, Mihaela Gaman, Andrei Avram, Mihai Ilie, Andrei Pruteanu, Adriana Stan, Lorena Rosia, Cristina Iacobescu, Luciana Morogan, George Dima, Gabriel Marchidan, Traian Rebedea, Madalina Chitez, Dani Yogatama, Sebastian Ruder, Radu Tudor Ionescu, Razvan Pascanu, and Viorica Patraucean. 2021. Liro: Benchmark and leaderboard for romanian language tasks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Erick Rocha Fonseca, Leandro Borges dos Santos, Marcelo Criscuolo, and Sandra Maria Aluísio. 2016. Visão geral da avaliação de similaridade semântica e inferência textual. *Linguamática*, 8(2):3–13.

Jiyeon Ham, Yo Joong Choe, Kyubyong Park, Ilji Choi, and Hyungjoon Soh. 2020. KorNLI and KorSTS:

New benchmark datasets for Korean natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 422–430, Online. Association for Computational Linguistics.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2101–2112, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hansi Hettiarachchi, Tharindu Ranasinghe, Paul Rayson, Ruslan Mitkov, Mohamed Gaber, Damith Premasiri, Fiona Anting Tan, and Lasitha Randunu Chandrakantha Uyangodage. 2025. Overview of the first workshop on language models for low-resource languages (LoResLM 2025). In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 1–8, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hiroki Iida and Naoaki Okazaki. 2021. Incorporating semantic textual similarity and lexical matching for information retrieval. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 582–591, Shanghai, China. Association for Computational Lingustics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Jcs Kadupitiya, Surangika Ranathunga, and Gihan Dias. 2016. Sinhala short sentence similarity calculation using corpus-based and knowledge-based similarity measures. In *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016)*, pages 44–53, Osaka, Japan. The COLING 2016 Organizing Committee.

Jenna Kanerva, Filip Ginter, Li-Hsin Chang, Iiro Rastas, Valtteri Skantsi, Jemina Kilpeläinen, Hanna-Mari Kupari, Jenna Saarni, Maija Sevón, and Otto Tarkka. 2021. Finnish paraphrase corpus. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 288–298, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. 2022. JGLUE: Japanese general language understanding evaluation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2957–2966, Marseille, France. European Language Resources Association.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chaofan Li, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Yingxia Shao, Defu Lian, and Zheng Liu. 2024. Making text embedders few-shot learners. *arXiv preprint arXiv:2409.15700*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Wanqin Ma, Chenyang Yang, and Christian Kästner. 2024. (why) is my prompt getting worse? rethinking regression testing for evolving llm apis. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI*, CAIN '24, page 166–171, New York, NY, USA. Association for Computing Machinery.

Manuel Mager, Elisabeth Mager, Alfonso Medina-Urrea, Ivan Vladimir Meza Ruiz, and Katharina Kann. 2018. Lost in translation: Analysis of information loss during machine translation between polysynthetic and fusional languages. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 73–83, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tamzeed Mahfuz, Satak Kumar Dey, Ruwad Naswan, Hasnaen Adil, Khondker Salman Sayeed, and Haz Sameen Shahgir. 2025. Too late to train, too early to use? a study on necessity and viability of

low-resource Bengali LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1183–1200, Abu Dhabi, UAE. Association for Computational Linguistics.

Goutam Majumder, Vikrant Rajput, Partha Pakray, Sivaji Bandyopadhyay, and Benoit Favre. 2024. Text summary evaluation based on interpretable semantic textual similarity. *Multimedia Tools and Applications*, 83(1):3233–3258.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Ruslan Mitkov, Le An Ha, Halyna Maslak, Tharindu Ranasinghe, and Vilelmini Sosoni. 2023. Automatic generation of multiple-choice test items from paragraphs using deep neural networks. In *Advancing Natural Language Processing in Educational Assessment*, pages 77–89. Routledge.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Wenchuan Mu and Kwan Hui Lim. 2024. Modelling text similarity: A survey. In *Proceedings of the 2023 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '23, page 698–705, New York, NY, USA. Association for Computing Machinery.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Satkunanantham Nilaxan and Surangika Ranathunga. 2021. Monolingual sentence similarity measurement using siamese neural networks for sinhala and tamil languages. In *2021 Moratuwa Engineering Research Conference (MERCon)*, pages 567–572.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Ji Yoon Han, Jangwon Park, Chisung Song, Junseong Kim, Youngsook Song, Taehwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Younghoon Jeong, Inkwon Lee, Sangwoo Seo, Dongjun Lee, Hyunwoo Kim, Myeonghwa Lee, Seongbo Jang, Seungwon Do, Sunkyoung Kim, Kyungtae Lim, Jongwon Lee, Kyumin Park, Jamin Shin, Seonghyun Kim, Lucy Park, Lucy Park, Alice Oh, Jung-Woo Ha (NAVER AI Lab), Kyunghyun Cho, and Kyunghyun Cho. 2021. Klue: Korean language understanding evaluation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Shenbin Qian, Archchana Sindhujan, Minnie Kabra, Diptesh Kanojia, Constantin Orasan, Tharindu Ranasinghe, and Fred Blain. 2024. What do large language models need for machine translation evaluation? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3660–3674, Miami, Florida, USA. Association for Computational Linguistics.

Tharindu Ranasinghe, Hansi Hettiarachchi, Nadeesha Pathirana, Damith Premasiri, Lasitha Uyangodage, Isuri Anuradha, Alistair Plum, Paul Rayson, and Ruslan Mitkov. 2025. Sinhala encoder-only language models and evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, Vienna, Austria. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2019a. Enhancing unsupervised sentence similarity methods with deep contextualised word representations. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 994–1003, Varna, Bulgaria. INCOMA Ltd.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2019b. Semantic textual similarity with Siamese neural networks. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1004–1011, Varna, Bulgaria. INCOMA Ltd.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Md Shajalal and Masaki Aono. 2018. Semantic textual similarity in bengali text. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–5.

Jakub Sido, Michal Seják, Ondřej Pražák, Miloslav Konopík, and Václav Moravec. 2024. Czech news dataset for semantic textual similarity. *Language Resources and Evaluation*, pages 1–18.

Vésteinn Snæbjarnarson, Annika Simonsen, Goran Glavaš, and Ivan Vulić. 2023. Transfer to a low-resource language via close relatives: The case study on Faroese. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 728–737, Tórshavn, Faroe Islands. University of Tartu Library.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

John Vidler and Paul Rayson. 2024. UCREL - Hex; a shared, hybrid multiprocessor system. https://github.com/UCREL/hex.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Yuxia Wang, Minghan Wang, and Preslav Nakov. 2024a. Rethinking STS and NLI in large language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 965–982, St. Julian's, Malta. Association for Computational Linguistics.

Zhimeng Wang, Pinzheng Wang, Juntao Li, Yibin Chen, and Min Zhang. 2024b. Achieving stronger generation via simple contrastive tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8986–8999, Miami, Florida, USA. Association for Computational Linguistics.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. 2019. Beyond BLEU: Training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy. Association for Computational Linguistics.

A Annotation Guidelines

All the datasets in MUSTS follow the annotation guidelines in Table 3.

5	*The two sentences are completely equivalent, as they mean the same thing.*
	The bird is bathing in the sink. Birdie is washing itself in the water basin.
4	*The two sentences are mostly equivalent, but some unimportant details differ.*
	Two boys on a couch are playing video games. Two boys are playing a video game.
3	*The two sentences are roughly equivalent, but some important information differs/missing.*
	John said he is considered a witness but not a suspect. "He is not a suspect anymore." John said.
2	*The two sentences are not equivalent, but share some details.*
	They flew out of the nest in groups. They flew into the nest together.
1	*The two sentences are not equivalent, but are on the same topic.*
	The woman is playing the violin. The young lady enjoys listening to the guitar.
0	*The two sentences are completely dissimilar.*
	The black dog is running through the snow. A race car driver is driving his car through the mud.

Table 3: Similarity scores with explanations and English examples from Agirre et al. (2013).

B Prompts

As a starting point, we used the prompt templates provided by Wang et al. (2024a), developed through a comprehensive experimental study, as the basis for our study. We made a few key updates. First, we revised the task description to explicitly mention 'semantic textual similarity' to avoid confusion with syntactic similarity, which our initial experiments suggested could occur when using just the term 'similarity'. We also added a new sentence to each prompt to clarify the expected output, aiming to reduce post-processing errors when extracting the scores. Table 4 summarises the final prompt templates.

C Supervised STS Models

C.1 Training Transformers

We trained the transformer models using the architecture in Figure 1.

- All our models use the AdamW (Loshchilov and Hutter, 2019) optimiser with a weight decay of 1e-8, learning rate of 2e-5, a warmup ratio of

task_description: Determine the semantic textual similarity between the following two sentences (S1, S2). The score should be ranging from 0.0 to 5.0, and can be a decimal.	
Strategy	**Prompt**
ZS	{task_description} Return the score only following the prefix 'Score:' without any other text or explanations. S1: {sentence1} S2: {sentence2}
FS	Five demonstration examples: Example 1: S1: Ebola UK: NHS staff 'panicked' after suspected Ebola cases. S2: UK says investigating 2 suspected MERS cases. Score: 0.0 Example 2: ... {task_description} Return the score only following the prefix 'Score:' without any other text or explanations. S1: {sentence1} S2: {sentence2}
FS-CoT	{task_description} Return the explanation and score only following the prefixes 'Explain:' and 'Score:' without any other text. Six demonstration examples with explanation for each: Example 1: S1: The black dog is running through the snow. S2: A race car driver is driving his car through the mud. Explain: S1 and S2 are completely dissimilar. Score: 0.0 Example 2: ... S1: {sentence1} S2: {sentence2}

Table 4: Prompt templates used for experiments

0.06 from the training data and are trained for five epochs with a batch size of 32 (gradient accumulation is applied when needed), and a maximum length of 512 tokens. The values of the hyper-parameters (including the number of training epochs) were set to fixed values in order to ensure consistency between all the models.

- All the models were evaluated while training using a development set that had one-fifth of the rows separated from the training set before the start of the training process.
- The best checkpoints were selected on the development set. We use the Spearman correlation as the checkpoint selection criterion.
- We trained our models on an NVidia L40 48G GPU. All models were trained with half-precision (fp16) using the default PyTorch implementation.

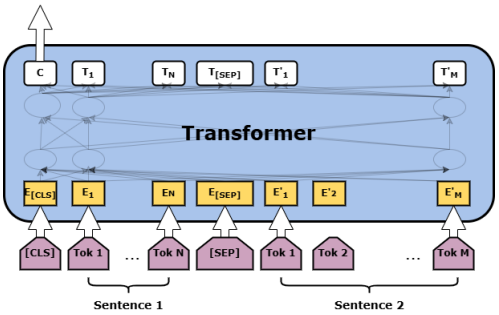


Figure 1: Architecture for using Transformers in STS.

C.2 Training LLM-based Encoders

We trained the LLM encoders using the architecture in Figure 2.

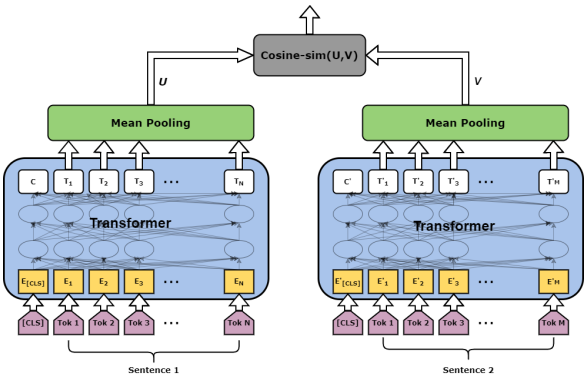


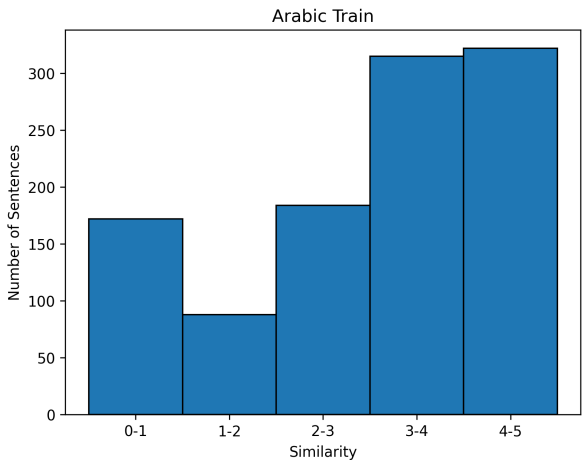
Figure 2: Architecture for using Transformers in STS.

- All models were trained using the same configurations mentioned in Appendix C.1, apart from the learning rate and warmup ratio, which were set to 1e-6 and 0.1, respectively.

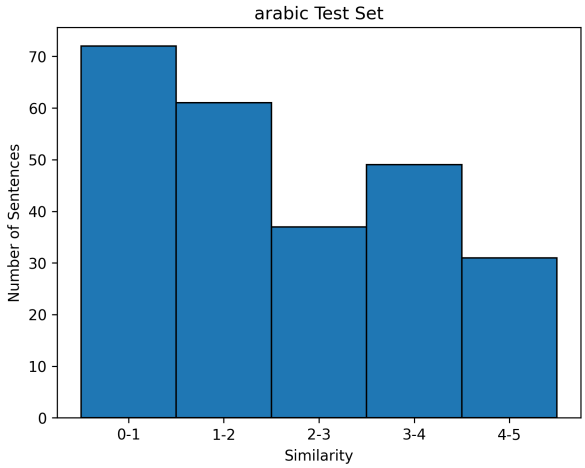
D MUSTS Statistics

In the following figures, we show label distribution and token overlap for each language.

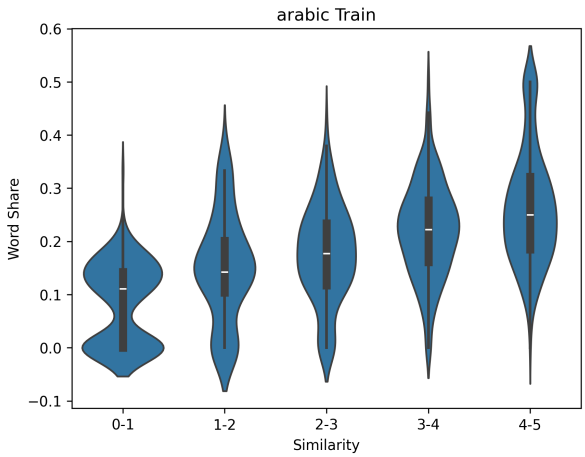
Arabic



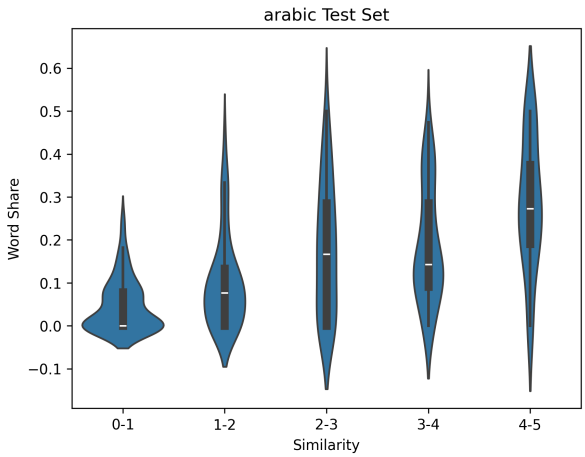
(1) Similarity distribution of Ar train set



(2) Similarity distribution of Ar test set

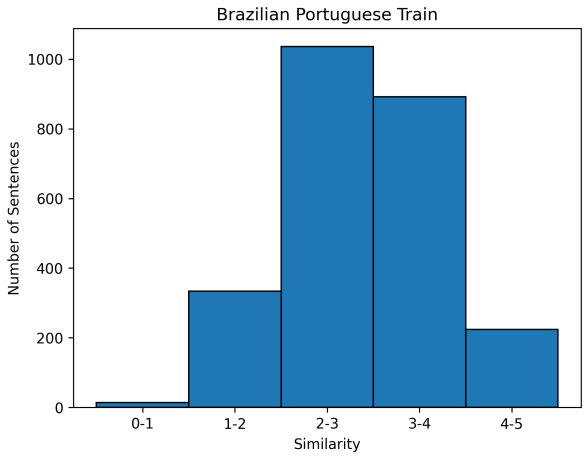


(3) Word share against similarity bins in Ar train set

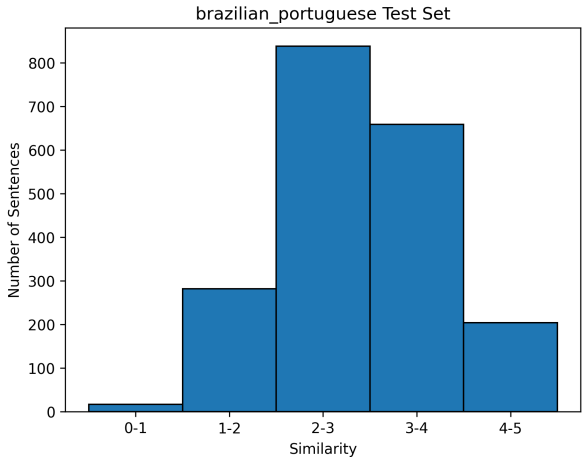


(4) Word share against similarity bins in Ar test set

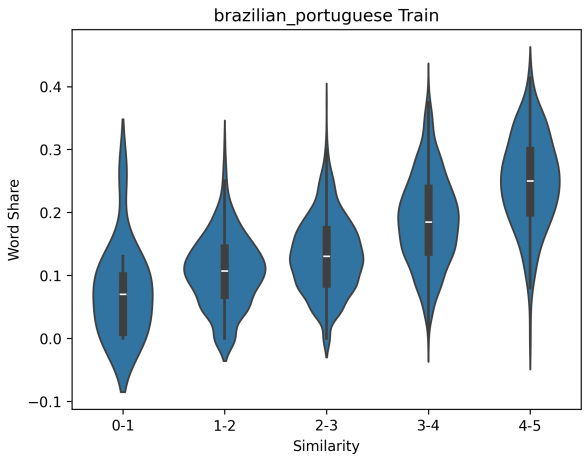
Brazilian Portuguese



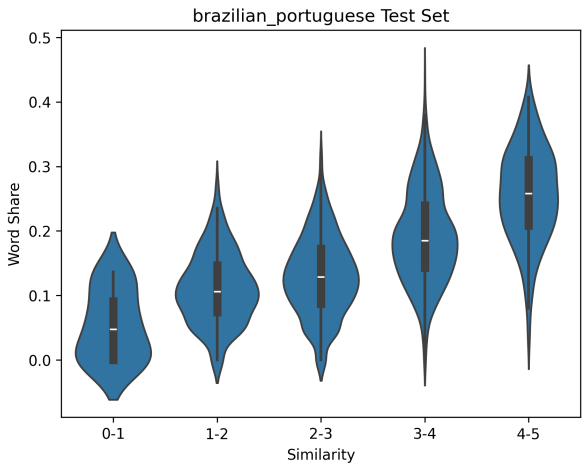
(1) Similarity distribution of Br-Pt train set



(2) Similarity distribution of Br-Pt test set

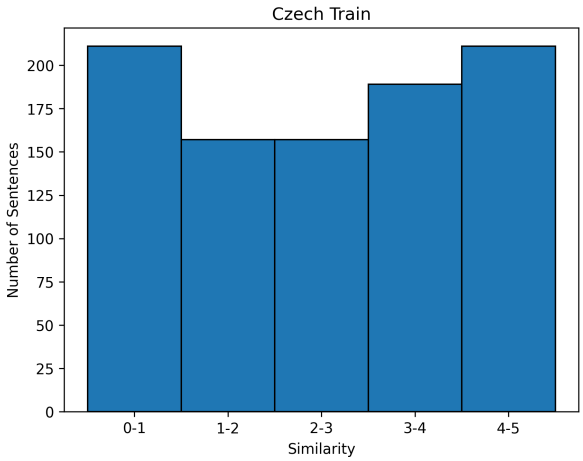


(3) Word share against similarity bins in Br-Pt train set

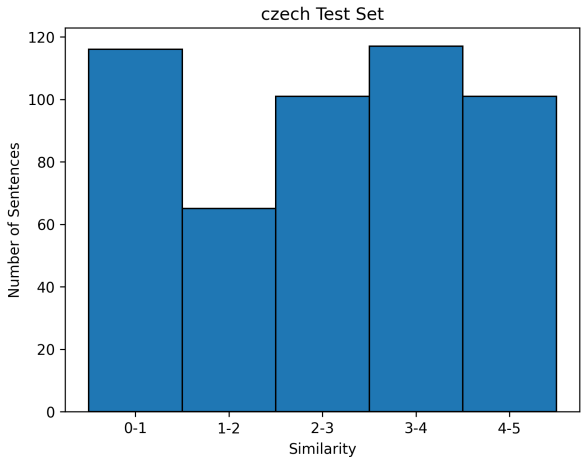


(4) Word share against similarity bins in Br-Pt test set

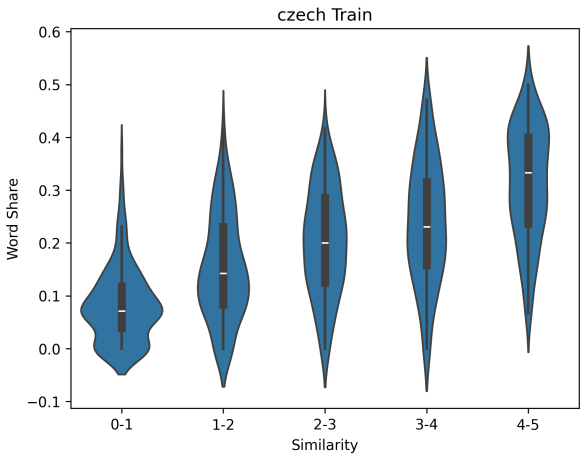
Czech



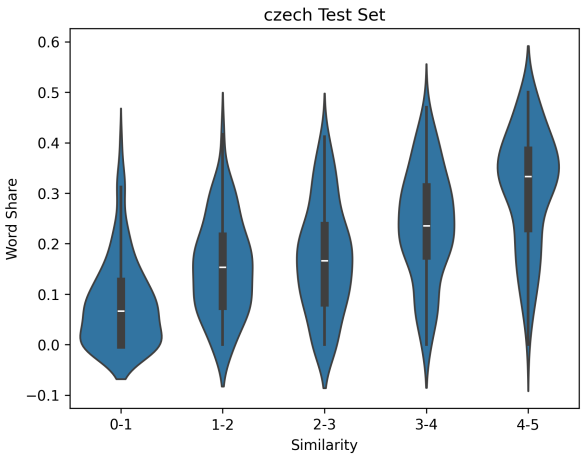
(1) Similarity distribution of Cz train set



(2) Similarity distribution of Cz test set

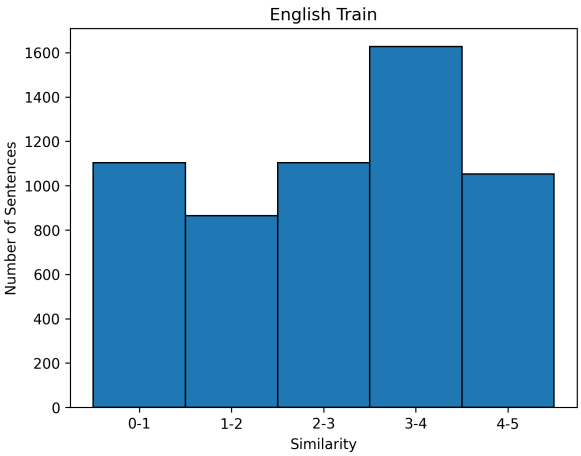


(3) Word share against similarity bins in Cz train set

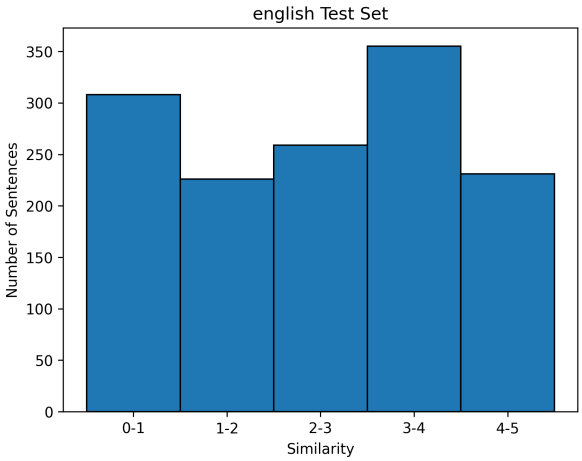


(4) Word share against similarity bins in Cz test set

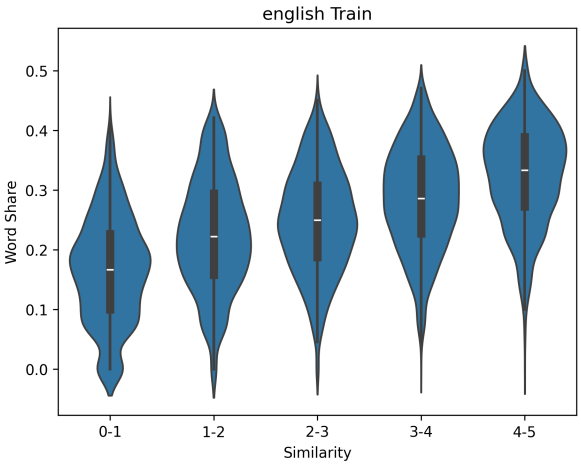
English



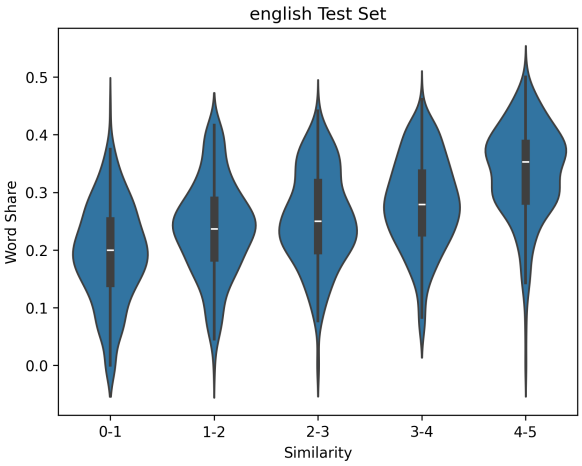
(1) Similarity distribution of En train set



(2) Similarity distribution of En test set

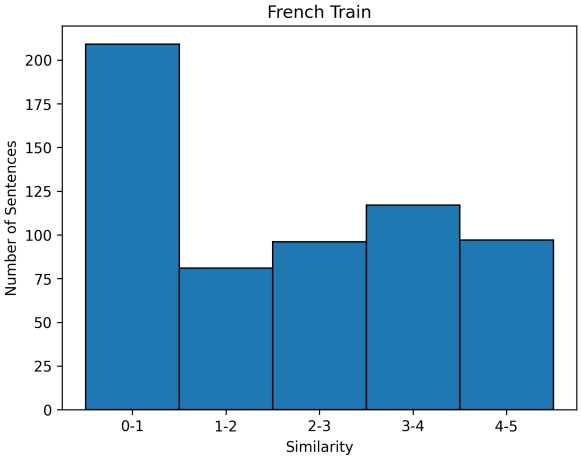


(3) Word share against similarity bins in En train set

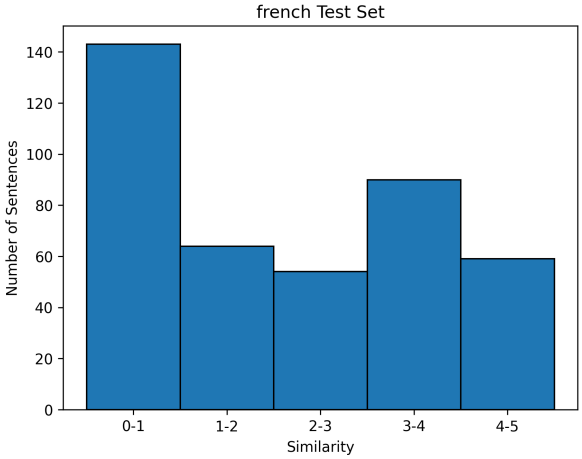


(4) Word share against similarity bins in En test set

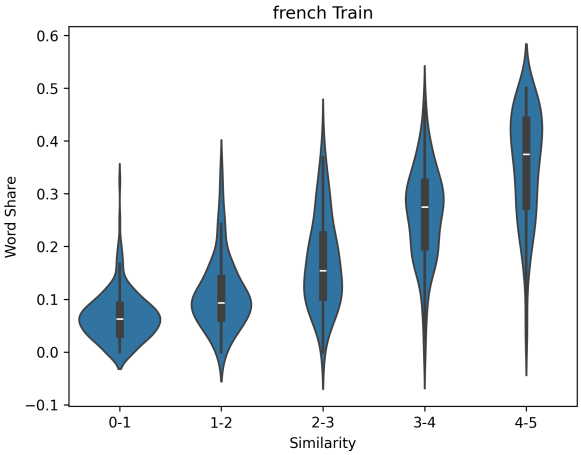
French



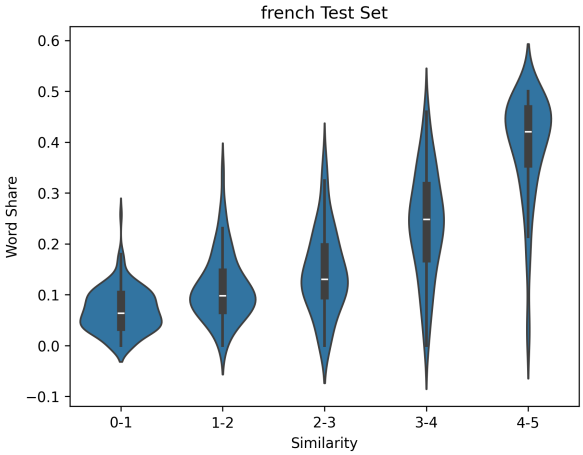
(1) Similarity distribution of Fr train set



(2) Similarity distribution of Fr test set

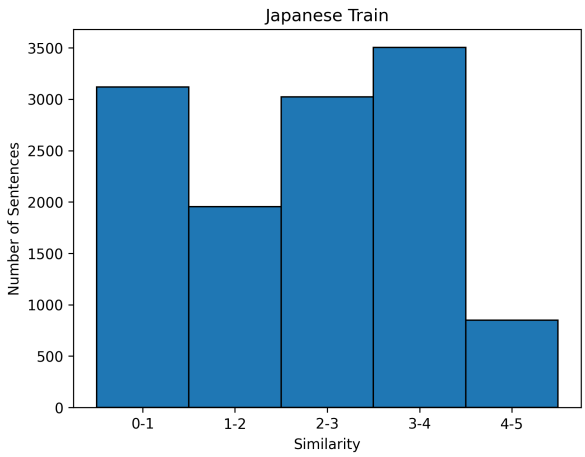


(3) Word share against similarity bins in Fr train set

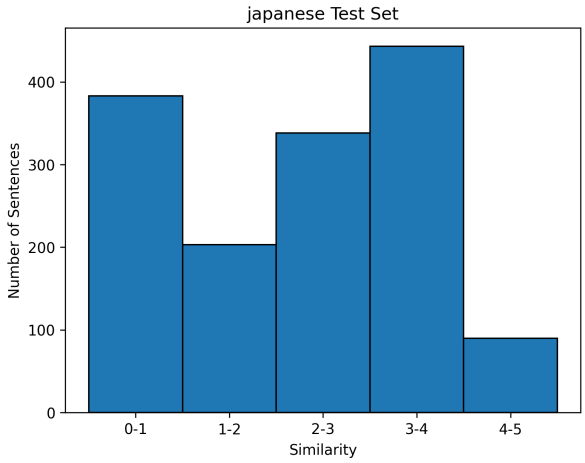


(4) Word share against similarity bins in Fr test set

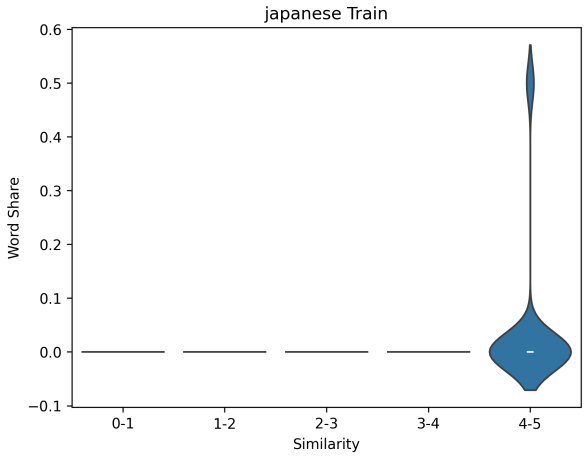
Japanese



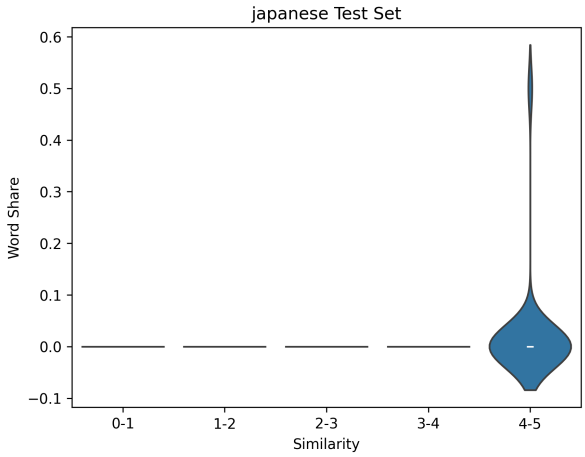
(5) Similarity distribution of Jp train set



(6) Similarity distribution of Jp test set

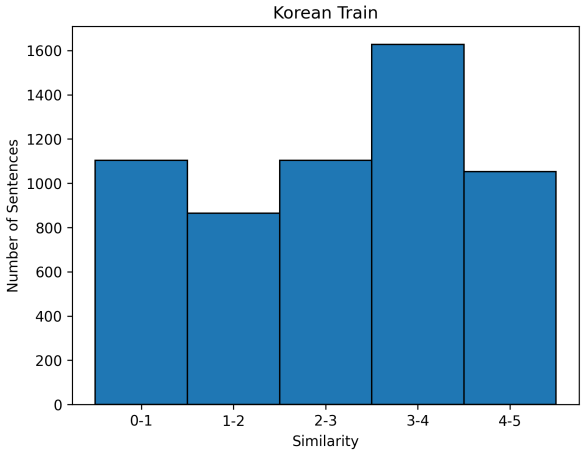


(7) Word share against similarity bins in Jp train set

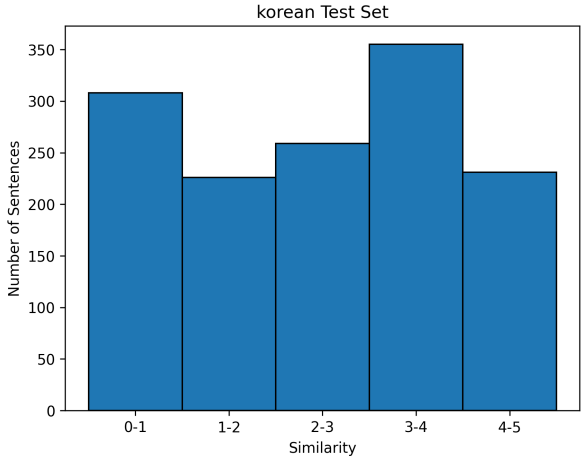


(8) Word share against similarity bins in Jp test set

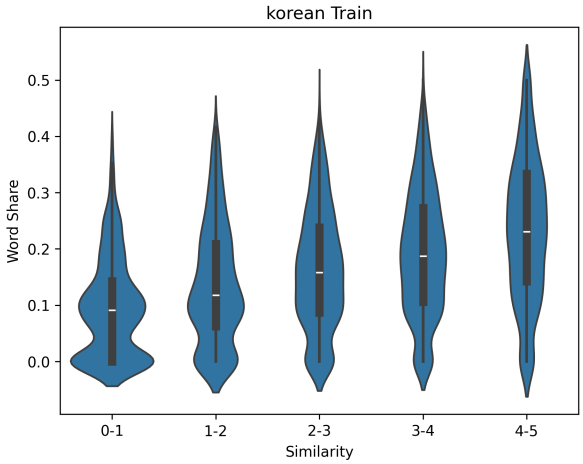
Korean



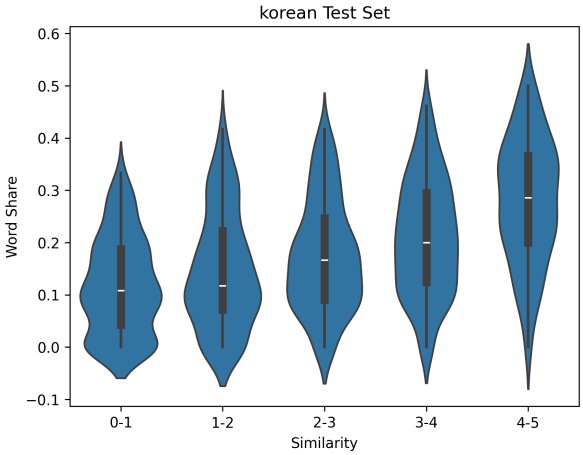
(1) Similarity distribution of Ko train set



(2) Similarity distribution of Ko test set

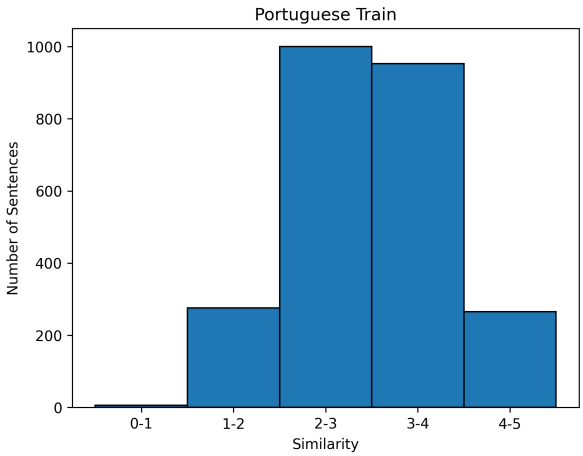


(3) Word share against similarity bins in Ko train set

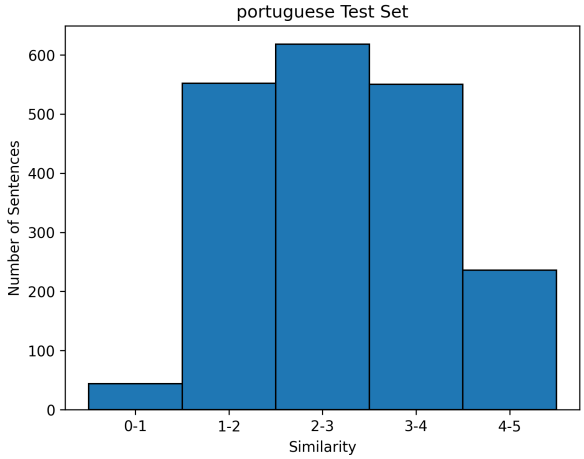


(4) Word share against similarity bins in Ko test set

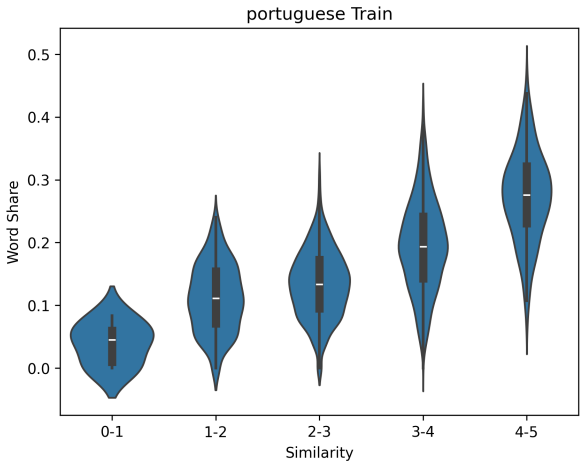
Portuguese



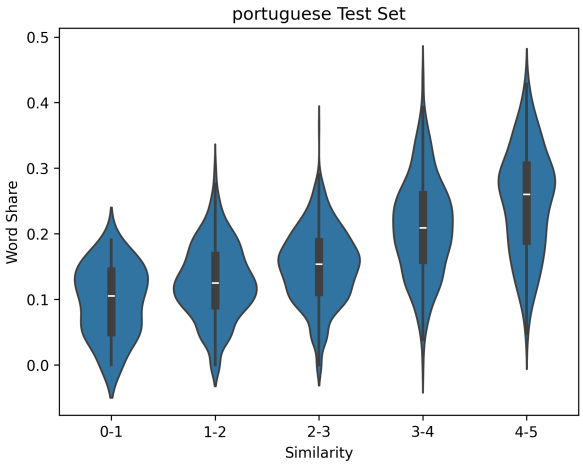
(1) Similarity distribution of Pt train set



(2) Similarity distribution of Pt test set

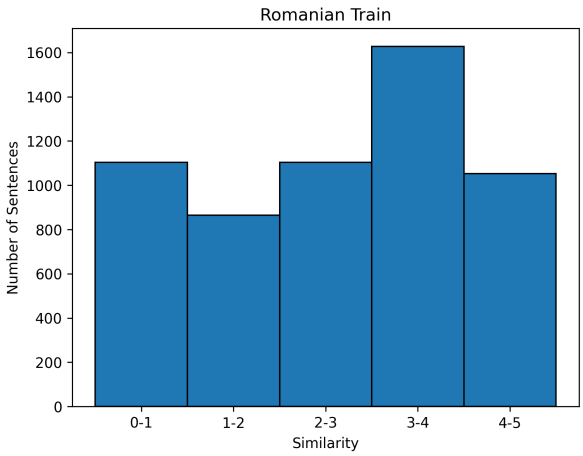


(3) Word share against similarity bins in Pt train set

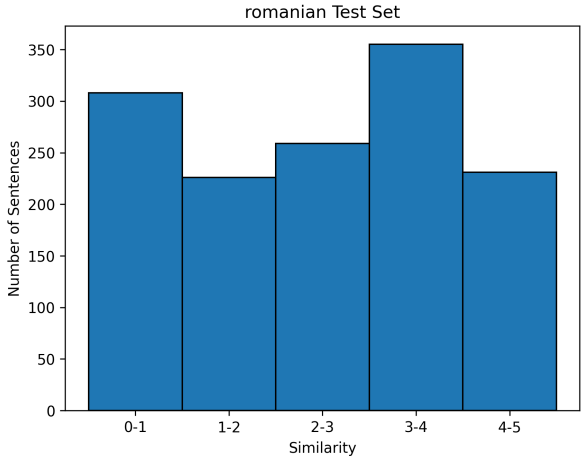


(4) Word share against similarity bins in Pt test set

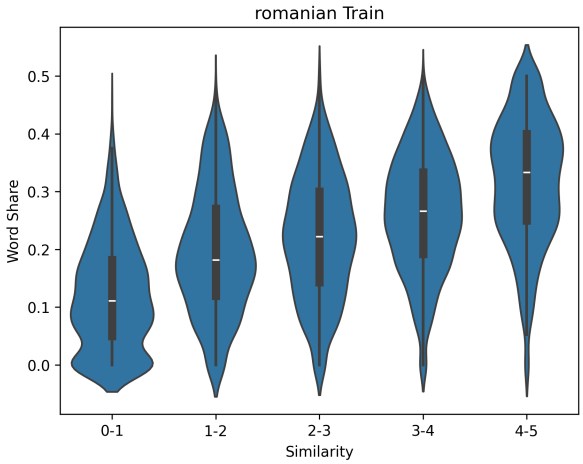
Romanian



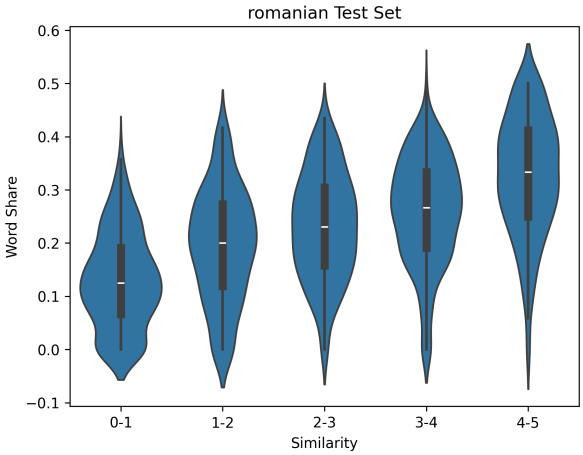
(1) Similarity distribution of Ro train set



(2) Similarity distribution of Ro test set

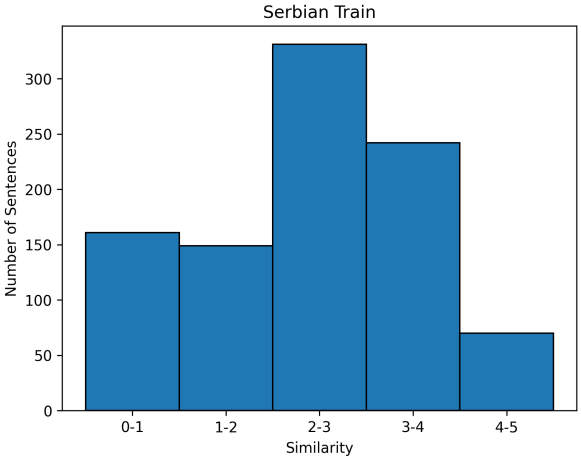


(3) Word share against similarity bins in Ro train set

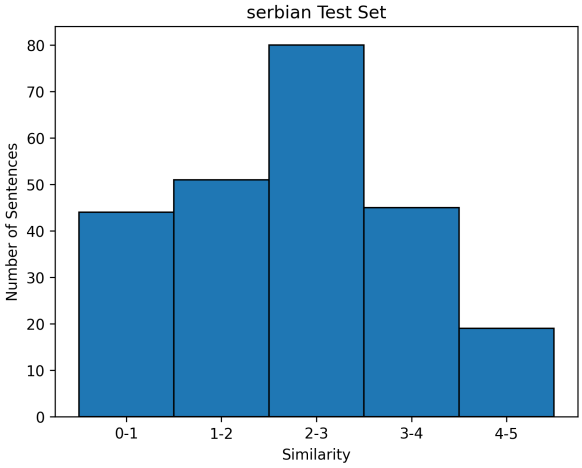


(4) Word share against similarity bins in Ro test set

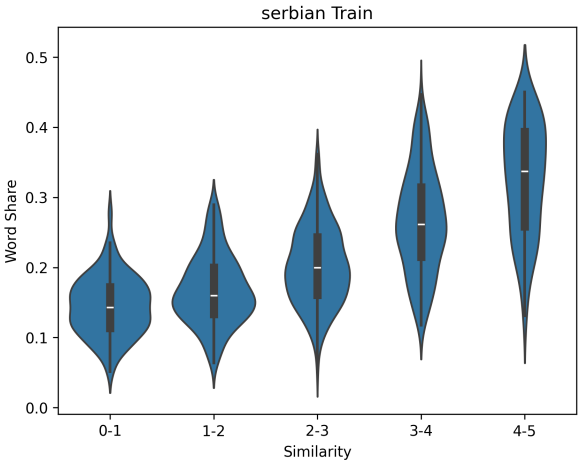
Serbian



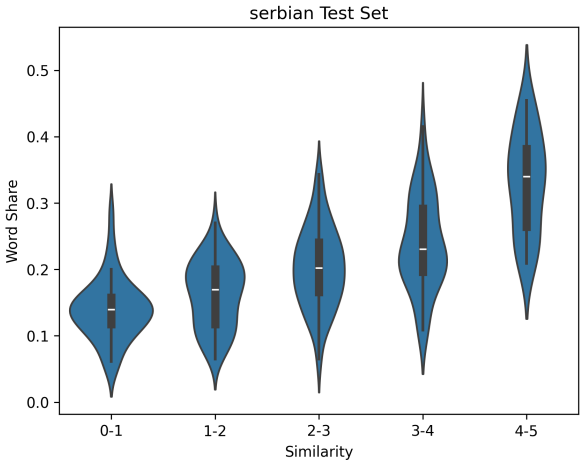
(1) Similarity distribution of Se train set



(2) Similarity distribution of Se test set

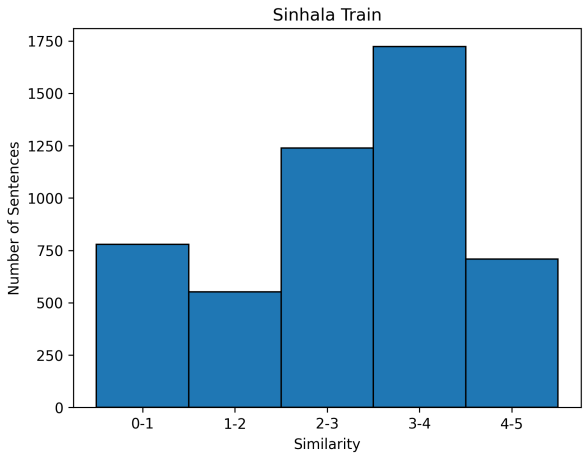


(3) Word share against similarity bins in Se train set

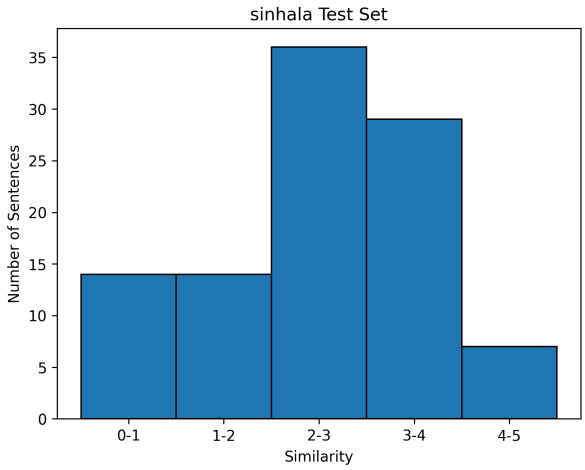


(4) Word share against similarity bins in Se test set

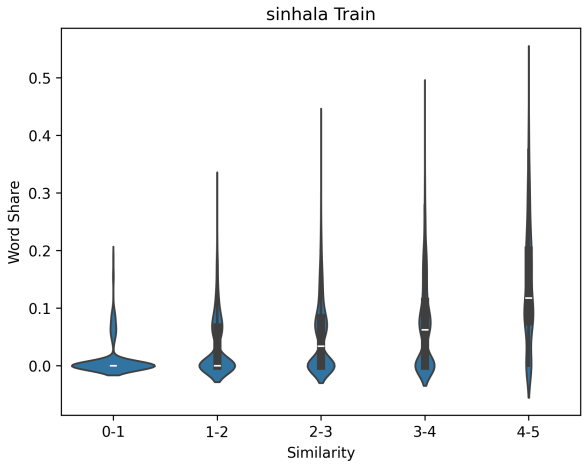
Sinhala



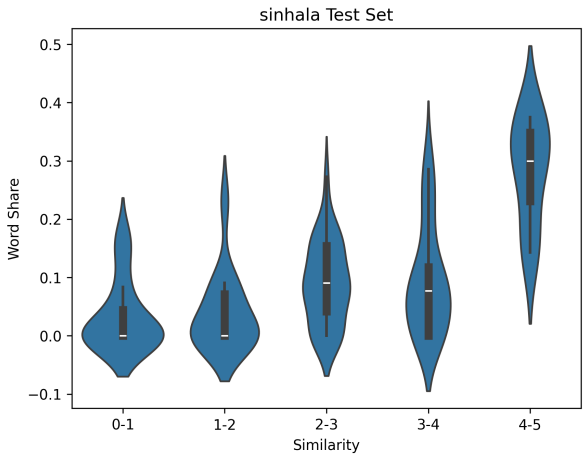
(1) Similarity distribution of Si train set



(2) Similarity distribution of Si test set

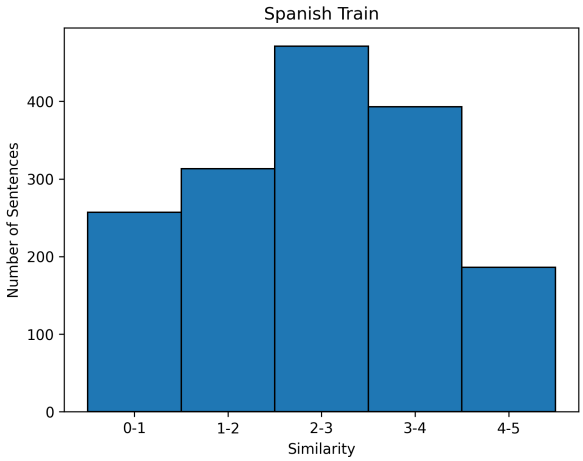


(3) Word share against similarity bins in Si train set

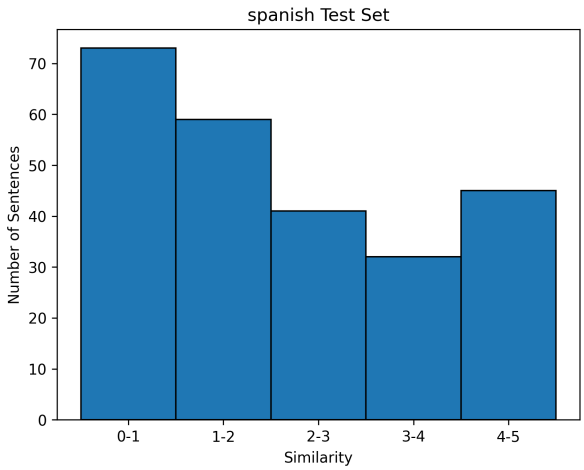


(4) Word share against similarity bins in Si test set

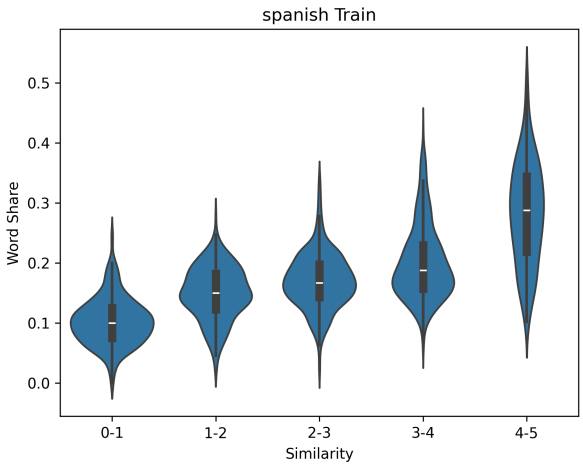
Spaish



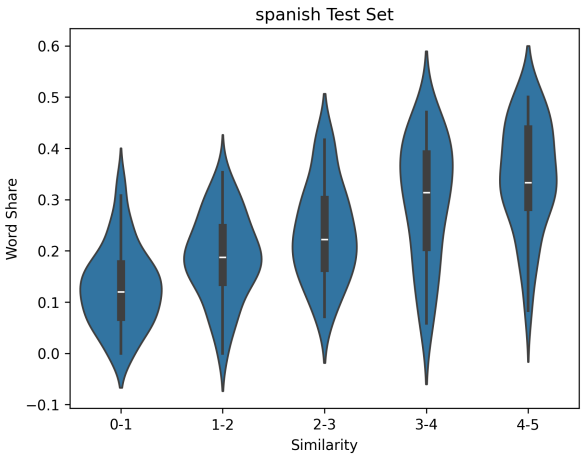
(1) Similarity distribution of Es train set



(2) Similarity distribution of Es test set

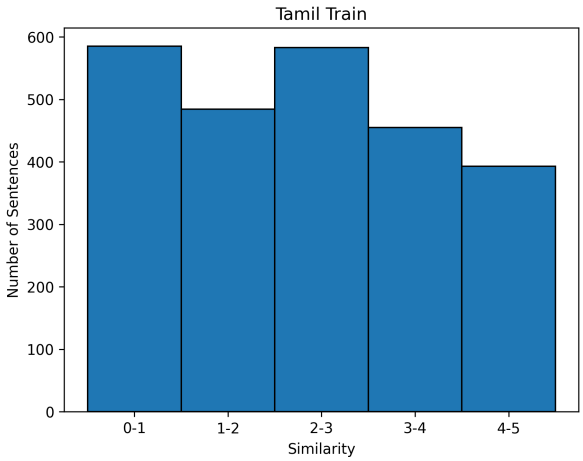


(3) Word share against similarity bins in Es train set

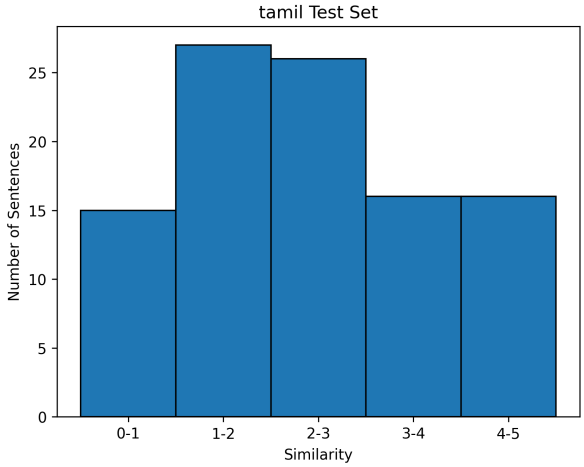


(4) Word share against similarity bins in Es test set

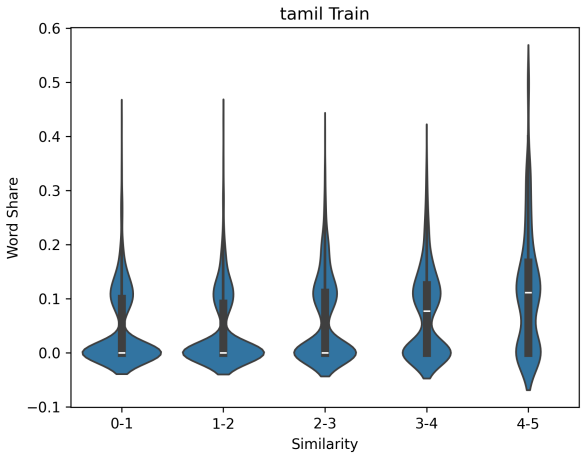
Tamil



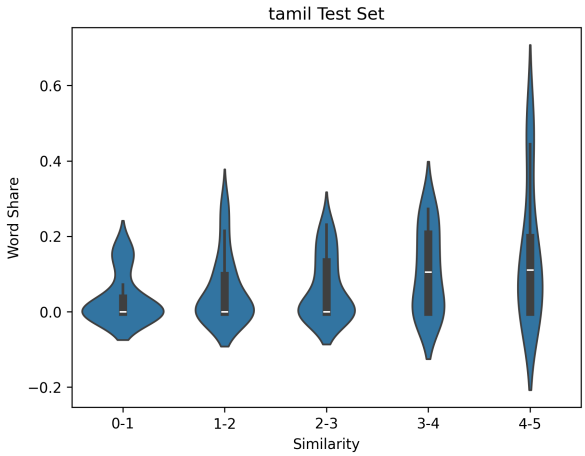
(1) Similarity distribution of Ta train set



(2) Similarity distribution of Ta test set



(3) Word share against similarity bins in Ta train set



(4) Word share against similarity bins in Ta test set

Can Large Language Models Accurately Generate Answer Keys for Health-related Questions?

Davis Bartles, Deepak Gupta, and Dina Demner-Fushman
National Library of Medicine, NIH
firstname.lastname@nih.gov

Abstract

The evaluation of text generated by large language models (LLMs) remains a challenge for question answering, retrieval augmented generation (RAG), summarization, and many other natural language processing tasks. Evaluating the factuality of LLM-generated responses is particularly important in medical question answering, where the stakes are high. One method of evaluating the factuality of text is through the use of information nuggets (answer keys). Nuggets are text representing atomic facts that may be used by an assessor to make a binary decision as to whether the fact represented by said nugget is contained in an answer. Given that manual nugget extraction is expensive and time-consuming, recent RAG shared task evaluations have explored automating the nuggetization of text with LLMs. In this work, we explore several approaches to nugget generation for medical question answering and evaluate their alignment with expert human nugget generation. We find providing an example and extracting nuggets from an answer to be the best approach to nuggetization. While, overall, we found the capabilities of LLMs to distill atomic facts limited, Llama 3.3 performed the best out of the models we tested.

1 Introduction

Evaluation of automatically generated answers is a major bottleneck in the development of question-answering approaches. Although a need for new evaluation approaches was noticed as soon as the system-generated answers became more complex and abstract (Chen et al., 2019), to date, there are no widely accepted evaluation metrics to approximate human judgments on the quality and other aspects of the generated answers. The TREC 2024 Retrieval Augmented Generation track evaluation (Pradeep et al., 2024) revisited the nugget-based evaluation originally developed for judging answers to definition questions in the 2003 question answering track (Voorhees, 2004). Briefly, a nugget-based evaluation is a two-step process. In the first step, the assessors create a list of atomic facts (nuggets) that must be present in an answer for the answer to be judged correct and complete. In the second step, the assessors manually map each statement in a system-generated answer to the nuggets. Various performance metrics may then be computed. For example, the TREC 2024 RAG track labeled nuggets as *supported*, *partially supported*, or *not supported* by the answer and then computed the system scores by summing the scores of all nuggets and dividing the sum by the number of nuggets. Most importantly, the evaluation has shown correlations between scores derived from an automatic nugget evaluation and a manual nugget evaluation. This supports the belief that LLMs can be used to support evaluations as long as LLMs are not generating the ground truth (Soboroff, 2024).

To that end, we explore various approaches to the first step of the evaluation - LLMs' ability to generate atomic factual statements (*cf.* Table 1). This is particularly important in medical question answering, where, although infrequently, as demonstrated by the results of the 2024 TREC BioGen track evaluation, generated answers may contain inappropriate and potentially harmful information (Gupta et al., 2024).

The contributions of this work are as follows:

- We manually generate nuggets for the 2024 BioGen track topics.
- We propose a series of automated nugget-generation approaches considering the question, answer, and relevant documents.
- We evaluated the capabilities of LLMs' to generate nuggets.

Related work: Nugget generation could be viewed as a form of outline generation in two different settings: 1) a model or a person generating the nuggets has access to a set of answers or docu-

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 354–368
July 27 - August 1, 2025

Query: *"What will mutation in runx2 affect in the future?"*
Answer: *"The effect of the runx2 mutation depends on the kind of the mutation. This gene mostly affects bone development. Mutations can cause bone deformities, height lower than expected, extra teeth and other dental problems..."*
Manually Extracted Nuggets:
1. Affects (runx2 mutation, bone development)
2. Cause (runx2 mutation, bone deformities)
3. Cause (runx2 mutation, height lower than expected)
4. Cause (runx2 mutation, extra teeth)
5. Cause (runx2 mutation, dental problems) ...
LLM (Llama 3.3) Generated Nuggets:
1. Runx2 mutation affects bone development
2. Runx2 mutation can cause bone deformities
3. Runx2 mutation can result in lower than ...
4. Runx2 mutation can lead to extra teeth
5. Runx2 mutation can cause other dental problems

Table 1: An example of nuggets extracted by a human and LLM for the same query and answer pair.

ments containing information needed to generate the answer; 2) the model or person are provided only with the question and have to generate the outline using their background knowledge. While, to the best of our knowledge, work on direct nugget generation is limited to the above RAG evaluation and an evaluation in which the initial set of test nuggets is generated using ChatGPT (Dietz, 2024; Farzi and Dietz, 2024), the related work on outline generation includes story generation (Wang and Kreminski, 2024) and natural language outline for code generation (Shi et al., 2024). For medical question answering, nugget-based evaluation was revisited in the evaluation of answers to questions about COVID-19 asked by patients and clinicians (Goodwin et al., 2022). The nuggets were generated manually in this evaluation. Other evaluations that leverage fact extraction were proposed for questions about biographies (Min et al., 2023) and medical question answering (Wang et al., 2024).

2 Methods

2.1 Manual Nugget Generation

For the purpose of having ground truths to evaluate LLM generated nuggets against, we provide expert-curated, manually generated nuggets for the 2024 BioGen track topics. Nuggets were captured from 40 ground truth answers. Each nugget was captured as a semantic triplet in Predicate (subject, object) form. Nuggets were identified by manually assessing the atomic facts represented in each sentence and their corresponding predicate, subject, and object. Some sentences may contain multiple atomic facts, for example, sentences comprising multiple phrases conjoined by a coordinating conjunction or lists. In such cases, the semantic triple is identified for each phrase or item in the list and recorded separately. Predicates were normalized across the dataset by mapping to a list of expert-curated predicates deemed to be complete in their coverage of the dataset and in conveying represented facts. Each medical concept contained in either the subject or object was associated with a Concept Unique Identifier (CUI) from the Unified Medical Language System (UMLS) (Lindberg et al., 1993). These associations were made by manually assessing the closest match, if any, from the UMLS Metathesaurus Browser. Some facts required more complex nugget structures including, but not limited to, "if, then" clauses and comparisons. These nuggets preserve the underlying logical structure from the answer. We generated a total of 498 nuggets from 40 question-answer pairs which has an average of 12.45 nuggets. Each nugget was reviewed by at least two reviewers.

2.2 LLM-based Nugget Generation

Model Architectures: We tested both popular open-source and proprietary models for nugget generation. The list of models includes Llama (Grattafiori et al., 2024), Gemma (Riviere et al., 2024), Mistral (Jiang et al., 2023), Phi (Abdin et al., 2024), Qwen (Qwen et al., 2025), Vicuna (Chiang et al., 2023), Falcon (Almazrouei et al., 2023), DeepSeek (DeepSeek-AI et al., 2025), GPT (OpenAI et al., 2024), Gemini (Team et al., 2023), and Claude[1]. For some families of models, we included both larger and smaller versions.

Generation Strategies: We developed extensive strategies to generate the nuggets by considering different inputs to the LLMs. Specifically, we used questions, reference answers, and cited documents provided for each assertion in the reference answer. We instructed the models to generate the appropriate nuggets. The detailed strategies are as follows:
(1) Question: In the first strategy, we only pro-

[1] https://www.anthropic.com/claude/sonnet

vide the question to the LLMs and instruct them to generate all pertinent nuggets that directly address the user's query. We started with the zero-shot approach and extended our experiments to the few-shot approach to enable in-context learning, where we provide an example question and corresponding nuggets in the prompt to direct the model toward better performance. We call this strategy Q_0 (zero-shot) and Q_1 (one-shot).

(2) Question + Answer: We aim to assess LLMs' capability of distilling nuggets from the ground-truth answers. We hypothesized that LLMs are expected to perform well in this setting and it can be considered an upper bound for the first strategy. Similar to the first strategy, we devise two strategies (zero and one-shot) and call them QA_0 and QA_1.

(3) Question + Documents: Following the success of the retrieval augmented generation (RAG) approach in BioGen (Gupta et al., 2024), we devised another strategy in which the relevant documents, along with the question, were passed as input to the LLMs. To get the relevant documents, we used the two-stage approach, in which we first used BM25 to retrieve the top 100 relevant documents from the BioGen 2024 PubMed corpus, and then re-ranked and selected the top 10 relevant documents using GraphMonoT5 approach (Gupta and Demner-Fushman, 2024). We aimed to investigate the role of input documents in the model's capability of refining the final nuggets. Toward this, we developed two variants of this approach. In the first variant, we feed all the retrieved documents together to the model, and in the second variant, we feed each document sequentially and instruct the model to refine the nuggets and produce the final nuggets at the end of the iteration. We call the former variant QRD_{all} and the latter QRD_{seq}. We also extended this strategy to the ground-truth documents and used the cited documents associated with each assertion in the reference answer. We call these variants QGD_{all} (all documents together) and QGD_{seq} (sequential documents).

(4) Question + Answer + Documents: Similar to the Question + Documents strategy, we devise other strategies where we include the ground-truth answer in the sequential processing of documents (QRD_{seq}, QGD_{seq}) and all documents together (QRD_{all}, QGD_{all}) settings, and call them ($QARD_{seq}$, $QAGD_{seq}$) and ($QARD_{all}$, $QAGD_{all}$) for sequential processing of documents and all documents together, respectively.

We have provided all the prompts and experimental details in the **Appendix**.

2.3 Evaluation Metrics

For a given question Q, and its ground-truth nuggets $Y = \{y_1, y_2, \ldots, y_m\}$ and model nuggets $X = \{x_1, x_2, \ldots, x_n\}$ of the size m and n respectively, we aim to match each nugget $X_i \in X$ to one of the ground-truth nuggets $y_j \in Y$. We formulate the nuggets matching as an assignment problem, where we first compute the semantic similarity $sim(x_i, y_j) = cosine(emb_{x_i}, emb_{y_j})$ between $x_i \in X$ and $y_j \in Y$ and create a similarity matrix $S \in \mathcal{R}^{m \times n}$. We then group all the elements of matrix S and sort them in descending order. Iteratively, we assign each x_i to y_j if, $S_{ij} >= \theta$, and x_i and y_j have not been assigned. We continue the process until all x_i (having $sim(.) >= \theta$) has been assigned. We keep track of each assigned y_j and ensure each y_j is mapped to at most one x_i while maximizing global similarity. Once the assignment is done, we compute precision $p = \frac{|X \cap Y|}{m}$ and recall $r = \frac{|X \cap Y|}{n}$, and F1-score, where $|X \cap Y|$ denotes the number of generated nuggets that match the ground-truth nuggets. For computing semantic similarity, we use the SentenceTransformer (Reimers and Gurevych, 2019) model[2].

3 Results and Discussion

Key Results: Table 2 shows the experimental results of multiple generation strategies at the optimal[3] value of threshold θ. For the **Question** strategy, the Llama 3.3 (70B) model obtained the maximum F1-score of 34.03% in zero-shot setting. On the **Question + Answer** strategy, which can be considered as an upper-bound for the LLMs, the Llama 3.3 (70B) model achieved a maximum F1-score of 76% in one-shot setting. Under **Question + Document** strategy, all the LLMs exhibited suboptimal performance and showed a maximum of 34.11% F1-score with the Gemini 2.0 Flash model, where all the relevant documents along with the question are provided as input to the model. On the **Question + Answer + Documents** strategy, the Qwen 2.5 (72B) model achieved a maximum F1-score of 62.95% where the relevant documents (one at a time until all the documents finished), along with

[2] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[3] The optimal value (0.7) was determined by manual comparison of 10 different sets of LLM and ground-truth nuggets.

Models	Question		Question + Answer		Question + Documents				Question + Answer + Documents			
	Q_0	Q_1	QA_0	QA_1	QGD_{all}	QRD_{all}	QGD_{seq}	QRD_{seq}	$QAGD_{all}$	$QARD_{all}$	$QAGD_{seq}$	$QARD_{seq}$
DeepSeek-R1 (7B)	17.28	19.11	34.07	53.61	9.86	7.95	16.91	5.82	9.68	8.85	13.3	5.61
DeepSeek-R1 (70B)	25.56	28.39	62.97	68.1	14.3	8.55	19.53	13.82	10.82	10.87	33.9	28.36
Falcon 3 (7B)	25.67	24.54	54.42	44.83	13.16	9.37	25.24	17.21	14.52	11.56	45.36	43.23
Falcon 3 (10B)	23.5	28.21	51.76	60.32	13.69	9.87	21.73	13.22	12.41	9.01	47.61	48.76
Gemma 2 (9B)	23.91	23.93	50.4	62.03	11.67	10.95	19.07	17.95	13.1	11.21	43.38	37.19
Gemma 2 (27B)	27.11	27.42	59.61	65.16	14.0	9.52	18.42	13.09	13.03	13.32	34.15	31.39
Llama 3.2 (3B)	19.15	15.52	37.14	50.32	13.6	11.56	18.36	11.37	15.17	8.82	39.52	41.57
Mistral Small (24B)	26.97	28.65	41.02	67.41	12.56	9.04	21.25	17.16	13.62	8.34	38.53	32.85
Phi-4 (14B)	26.57	26.32	61.84	66.33	12.68	9.23	26.29	15.68	12.4	10.98	39.03	37.11
Qwen2.5 (7B)	22.43	25.96	64.38	65.2	11.24	7.86	20.43	12.56	11.21	9.13	34.11	25.69
Qwen2.5 (72B)	28.39	34.52	67.45	72.68	10.96	8.64	29.34	**24.39**	12.3	9.31	**56.41**	**62.95**
Vicuna1.5 (7B)	17.71	21.13	53.54	41.12	7.3	4.27	13.43	12.4	7.72	6.32	36.31	36.33
Vicuna1.3 (33B)	18.48	23.15	55.24	60.63	8.15	5.07	15.93	9.61	7.33	6.32	32.53	23.08
Llama 3.3 (70B)	**34.03**	**33.45**	**68.32**	**76**	17.53	10.94	**29.37**	22	18.29	12.72	39.1	44.8
GPT-4o	33.48	31.22	64.03	69.82	29.63	**20.16**	24.55	21.17	7.63	10.2	24.39	44.17
Gemini 2.0 Flash	33.14	35.32	56.05	72.55	**34.11**	15.29	8.81	19.85	50.83	43.68	32.64	31.02
Claude 3.5 Sonnet	27.08	16.3	66.11	67.86	33.17	17.44	17.16	16.56	**51.48**	**43.82**	47.87	46.18

Table 2: Performance comparison of various open and closed-source LLMs on the task of nugget generation under different generation strategies. All the results are reported here denote the F1-score.

the question and answer, are provided as input to the model.

Discussion and Findings: We observed a **significant performance gap among the generation strategies**. The best model's performance difference between **Question + Answer** and **Question** is 41.97%. Similarly, we recorded a performance difference between **Question + Answer** and **Question + Document** as 41.89%. With the ground-truth documents as well, the GPT-4o obtained an F1-score of 29.63% compared to its counterpart **Question** strategy with an F1-score of 33.48%. Similar observations are made for most of the open-source LLMs, except for the Gemini 2.0 Flash model, where the difference between **Question + Documents** (34.11) and **Question** (33.14) strategy is not significant.

We observed that smaller models (3B-14B) tend to obtain lower performance compared to their counterpart larger models. The study also reveals that **LLMs lack the capability of accurately generating or extracting nuggets** for the health-related **query**. Table 2 exhibits the performance of **Question** strategy that tests the LLMs' knowledge in generating the nuggets for the given question, which does not achieve the anticipated performance. For the **Question + Answer** strategy, where the ground-truth answer was given to the model to extract the nuggets, it only achieves the F1-score of 76% which highlights LLMs limitation in accurately distilling the atomic facts from the answer. LLMs showed similar behavior when the documents were given to the model for generating/extracting the nuggets.

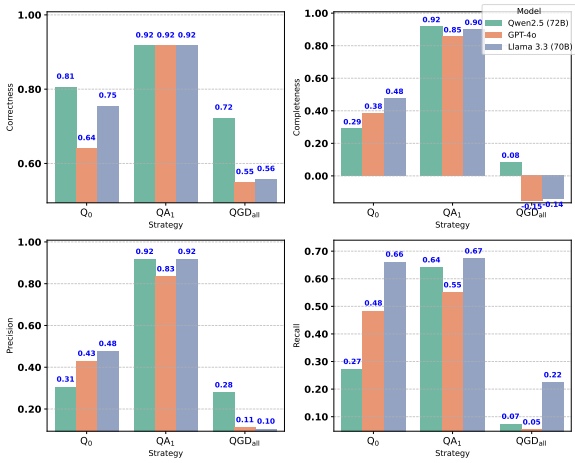


Figure 1: Performance comparison of different models under different generation strategies on multiple human evaluation criteria.

Human Evaluation: We also performed an extensive human analysis on the LLM-generated nuggets on multiple evaluation criteria. For the human evaluation, we chose the top-3 best-performing LLMs across multiple settings. We evaluated a total of 320 model-generated nuggets with 37 ground-truth nuggets for three diverse questions across three different settings: Q_0, QA_1, and QGD_{all}. We evaluate the quality of each nugget on the following criteria: **(a) Correctness**: whether the generated nugget is correct (2), partially correct (1), and incorrect (0); **(b) Completeness**: whether the generated nugget is misleading (-1), not required (0), Okay, but not required (1), and required (2); **(c) Precision**: portions of the generated nuggets that are correct; **(d) Recall**: portions of the ground-truth nuggets covered in the generated nuggets. We computed all the aforemen-

tioned scores for each question and averaged them to report (Fig. 1) the overall scores. The human evaluation of completeness and correctness criteria reveals that under the ground-truth answers (QA_1 strategy) all three LLMs' performance was better, only the question (Q_0) strategy obtained the sub-optimal performance. The evaluation also highlights that automatic precision and recall are highly aligned with manual precision and recall.

4 Conclusions

This work presented a comprehensive study on generating nuggets for health-related questions using various open and closed-source LLMs. Firstly, we manually formulated nuggets for BioGen 2024 topics and thereafter, we devised multiple nugget generation strategies to assess the capability of LLMs under different settings. We found that most LLMs obtained sub-optimal performance on the task which demonstrates the challenge involved with nugget generation, and we believe that our manual-created nuggets will promote further research in this direction.

Ethics Statement

The health-related questions and reference answers used in this study are publicly available within the Text REtrieval Conference (TREC) 2024 data.

Limitations

While the BioGen 2024 dataset covers a broad range of question topics and intents sourced out of popular health-related searches, it is not exhaustive. Subsequently, our findings on the LLMs' ability to generate nuggets in zero-shot settings apply to information needs covered in the data: clinical decision-support, factoid, and treatment and environment effects. The manual nugget evaluation approach outlined above can be used in the future to expand the data.

Another limitation of the data is a single reference answer. While the bulk of the nuggets must be present in any answer, some of the automatically generated nuggets could have been present in alternative answers. For example, if the existing reference answer list surgery as a treatment option, without specifying the best procedures, automatically generated nuggets that name specific surgeries will not get any credits for these nuggets. While this may somewhat lower the scores, it should not affect the model ranking, as the same approach is used for all models. In the future, more than one reference answer would be desirable to base the evaluation on a nugget pyramid (Marton and Radul, 2006).

Acknowledgments

This research was supported by the Division of Intramural Research (DIR) of the National Library of Medicine (NLM), National Institutes of Health. This work utilized the computational resources of the NIH HPC Biowulf cluster (https://hpc.nih.gov).

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *Preprint*, arXiv:2311.16867.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating question answering evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 119–124, Hong Kong, China. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai

Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Laura Dietz. 2024. A workbench for autograding retrieve/generate systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 1963–1972, New York, NY, USA. Association for Computing Machinery.

Naghmeh Farzi and Laura Dietz. 2024. Pencils down! automatic rubric-based evaluation of retrieve/generate systems. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '24, page 175–184, New York, NY, USA. Association for Computing Machinery.

Travis R Goodwin, Dina Demner-Fushman, Kyle Lo, Lucy Lu Wang, Hoa T Dang, and Ian M Soboroff. 2022. Automatic question answering for multiple stakeholders, the epidemic question answering dataset. *Scientific Data*, 9(1):432.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whit-

ney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Deepak Gupta and Dina Demner-Fushman. 2024. Empowering language model with guided knowledge fusion for biomedical document re-ranking. In *International Conference on Artificial Intelligence in Medicine*, pages 251–260. Springer.

Deepak Gupta, Dina Demner-Fushman, William Hersh, Steven Bedrick, and Kirk Roberts. 2024. Overview of trec 2024 biomedical generative retrieval (biogen) track. *Preprint*, arXiv:2411.18069.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Donald AB Lindberg, Betsy L Humphreys, and Alexa T

McCray. 1993. The unified medical language system. *Yearbook of medical informatics*, 2(01):41–51.

Gregory A. Marton and Alexey Radul. 2006. Nuggeteer: Automatic nugget-based evaluation using descriptions and judgements. In *North American Chapter of the Association for Computational Linguistics*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay,

Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. 2024. Initial nugget evaluation results for the trec 2024 rag track with the autonuggetizer framework. *arXiv preprint arXiv:2411.09607*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Gemma Team Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, L'eonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ram'e, Johan Ferret, Peter Liu, Pouya Dehghani Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stańczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Boxi Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Christoper A. Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozi'nska, D. Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Pluci'nska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost R. van Amersfoort, Josh Gordon, Josh Lipschultz, Joshua Newlan, Junsong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, L. Sifre, Lena Heuermann, Leti cia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Gorner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Peng chong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Perrin, S'ebastien M. R. Arnold, Se bastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomás Kociský, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeffrey Dean, Demis Hassabis, Koray Kavukcuoglu, Clément Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv*, abs/2408.00118.

Kensen Shi, Deniz Altınbüken, Saswat Anand, Mihai Christodorescu, Katja Grünwedel, Alexa Koenings, Sai Naidu, Anurag Pathak, Marc Rasi, Fredde Ribeiro, et al. 2024. Natural language outlines for code: Literate programming in the llm era. *arXiv preprint arXiv:2408.04820*.

Ian Soboroff. 2024. Don't use llms to make relevance judgments. *Preprint*, arXiv:2409.15133.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Ellen M. Voorhees. 2004. Overview of the trec 2003 question answering track. In *Text Retrieval Conference*.

Huimin Wang, Yutian Zhao, Xian Wu, and Yefeng Zheng. 2024. imapScore: Medical fact evaluation made easy. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10242–10257, Bangkok, Thailand. Association for Computational Linguistics.

Phoebe J Wang and Max Kreminski. 2024. Guiding and diversifying llm-based story generation via answer set programming. *arXiv preprint arXiv:2406.00554*.

A Models

Table 3 is an exhaustive list of the models tested in our experiments with their versions and approximate number of parameters.

Model	Version / Size
Llama 3.2	3B[4]
Llama 3.3	70B[5]
Gemma 2	9B[6], 27B[7]
Mistral Small	24B[8]
Phi-4	14B[9]
Qwen2.5	7B[10], 72B[11]
$\text{Vicuna}_{1.5}$	7B[12]
$\text{Vicuna}_{1.3}$	33B[13]
Falcon 3	7B[14], 10B[15]
DeepSeek-R1	7B[16], 70B[17]
GPT-4o	`gpt-4o-2024-08-06`[18]
Claude 3.5 Sonnet	`claude-3-5-sonnet-20240620`[19]
Gemini 2.0 Flash	`gemini-2.0-flash`[20]

Table 3: A list of the models tested in our experiments.

[4] https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct
[5] https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct
[6] https://huggingface.co/google/gemma-2-9b-it
[7] https://huggingface.co/google/gemma-2-27b-it
[8] https://huggingface.co/mistralai/Mistral-Small-24B-Instruct-2501
[9] https://huggingface.co/microsoft/phi-4
[10] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct
[11] https://huggingface.co/Qwen/Qwen2.5-72B-Instruct
[12] https://huggingface.co/lmsys/vicuna-7b-v1.5
[13] https://huggingface.co/lmsys/vicuna-33b-v1.3
[14] https://huggingface.co/tiiuae/Falcon3-7B-Instruct
[15] https://huggingface.co/tiiuae/Falcon3-10B-Instruct
[16] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B
[17] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B
[18] https://openai.com/index/hello-gpt-4o/
[19] https://www.anthropic.com/claude/sonnet
[20] https://deepmind.google/models/gemini/flash/

B Details of Experimental Setup

Details of LLM Prompting To test the capabilities of LLMs to generate and extract nuggets, we prompted each model with a series of approaches. Table 4 contains the prompt used for Q_0, the zero-shot variation of our **Question** strategy. Table 5 contains the prompt used for QA_0, the zero-shot variation of our **Question + Answer** strategy. Table 6 contains the prompt used for Q_1, the one-shot variation of our **Question** strategy. Table 7 contains the prompt for QA_1, the one-shot variation of our **Question + Answer** strategy. Table 8 contains the prompt for QGD_{all} and QRD_{all}, our **Question + Document** strategies with all ground-truth documents and all retrieved documents, respectively. Table 9 contains the prompt for QRD_{seq} and QGD_{seq}, our **Question + Document** strategies with sequential ground-truth documents and sequentially retrieved documents, respectively. Table 10 contains the prompt for $QAGD_{all}$ and $QARD_{all}$, our **Question + Answer + Document** strategies with all ground-truth documents and all retrieved documents, respectively. Table 11 contains the prompt for $QARD_{seq}$ and $QAGD_{seq}$, our **Question + Answer + Document** strategies with sequential ground-truth documents and sequentially retrieved documents, respectively.

Each prompt contains some combination of query (q), answer (a), context (c), and initial nugget list (i) variables. The query and answer variables were substituted with each query and answer from the BioGen 2024 dataset. For the settings with all documents, the context variable was substituted with a list of the abstracts from all documents for the query separated by a new line character. For the settings with sequential documents, the context variable was substituted with a single abstract. The prompt for the sequential documents settings also contained the initial nugget list variable. This variable was initially "None" and then was substituted with the list of nuggets produced by the model provided with the previous abstract. For the sequential documents settings, the models were prompted once with each abstract and only the final list of nuggets was recorded. All models were prompted with their default settings (e.g. temperature).

SYSTEM: You are NuggetGenerateLLM, an AI assistant specialized in generating all information nuggets that are required to completely answer a given query. A nugget is an atomic fact.
USER: Generate all the information nuggets that are required to completely answer the query given below. Each nugget must contain one, and only one, fact. A nugget must be as concise and as specific as possible. A nugget cannot contain a list, each element in a list must be its own nugget. Each nugget must directly answer the query. The list of nuggets must not contain redundant information. Return a list of nuggets such that each nugget is on a new line. Do not number or bullet the list. Do not include anything in your response except for the list of nuggets. Here is an example of the output format:
nugget1
nugget2
...
Query: q
LLM:
nugget1
nugget2
...

Table 4: Prompt for Q_0.

SYSTEM: You are NuggetExtractLLM, an AI assistant specialized in extracting information nuggets from a given answer. A nugget is an atomic fact.
USER: Generate all the information nuggets that are required to completely answer the query given below. Each nugget must contain one, and only one, fact. A nugget must be as concise and as specific as possible. A nugget cannot contain a list, each element in a list must be its own nugget. Each nugget must directly answer the query. The list of nuggets must not contain redundant information. Return a list of nuggets such that each nugget is on a new line. Do not number or bullet the list. Do not include anything in your response except for the list of nuggets. Here is an example of the output format:
nugget1
nugget2
...
Query: q
Answer: a
LLM:
nugget1
nugget2
...

Table 5: Prompt for QA_0.

SYSTEM: You are NuggetGenerateLLM, an AI assistant specialized in generating all information nuggets that are required to completely answer a given query. A nugget is an atomic fact.
USER: Generate all the information nuggets that are required to completely answer the query given below. Each nugget must contain one, and only one, fact. A nugget must be as concise and as specific as possible. A nugget cannot contain a list, each element in a list must be its own nugget. Each nugget must directly answer the query. The list of nuggets must not contain redundant information. Return a list of nuggets such that each nugget is on a new line. Do not number or bullet the list. Do not include anything in your response except for the list of nuggets. Here is an example of the output format:
nugget1
nugget2
...
Here is an example query: Why is transferrin and iron low in covid patients but ferritin high?
This is the list of nuggets that should be generated for this query:
Lymphocytes and viruses compete for iron.
Lymphocytes need iron for cellular response.
Lymphocytes need iron for humoral response.
Viruses need iron to replicate.
Infection lowers iron levels in the blood.
Infection increases ferritin levels in the blood.
High ferritin is associated with increased mortality.
Iron homeostasis needs ferritin.
Ferritin is involved in physiologic processes.
Ferritin is involved in pathologic processes.
High ferritin indicates response to inflammation.
High ferritin levels are linked to poor outcomes of COVID-19.
Iron depletion therapy showed anti-viral activity in the COVID-19 pandemic.
Iron depletion therapy showed anti-fibrotic activity in the COVID-19 pandemic.
Query: q
LLM:
nugget1
nugget2
...

Table 6: Prompt for Q_1.

SYSTEM: You are NuggetExtractLLM, an AI assistant specialized in extracting information nuggets from a given answer. A nugget is an atomic fact.
USER: List all of the information nuggets in the answer given below that are required to completely answer the query. Each nugget must contain one, and only one, fact from the answer. A nugget must be as concise and as specific as possible. Each element in a list must be its own nugget. Each nugget must directly answer the query. The list of nuggets must not contain redundant information. Return a list of nuggets such that each nugget is on a new line. Do not number or bullet the list. Do not include anything in your response except for the list of nuggets. Here is an example of the output format:
nugget1
nugget2
...
Here is an example query: Why is transferrin and iron low in covid patients but ferritin high?
This is the list of nuggets that should be generated for this query:
Lymphocytes and viruses compete for iron.
Lymphocytes need iron for cellular response.
Lymphocytes need iron for humoral response.
Viruses need iron to replicate.
Infection lowers iron levels in the blood.
Infection increases ferritin levels in the blood.
High ferritin is associated with increased mortality.
Iron homeostasis needs ferritin.
Ferritin is involved in physiologic processes.
Ferritin is involved in pathologic processes.
High ferritin indicates response to inflammation.
High ferritin levels are linked to poor outcomes of COVID-19.
Iron depletion therapy showed anti-viral activity in the COVID-19 pandemic.
Iron depletion therapy showed anti-fibrotic activity in the COVID-19 pandemic.
Query: q
Answer: a
LLM:
nugget1
nugget2
...

Table 7: Prompt for QA_1.

SYSTEM: You are NuggetGenerateLLM, an AI assistant specialized in using context to generate all information nuggets that are required to completely answer a given query. A nugget is an atomic fact.
USER: Use the context provided to generate all the information nuggets that are required to completely answer the query given below. Each nugget must contain one, and only one, fact. A nugget must be as concise and as specific as possible. A nugget cannot contain a list, each element in a list must be its own nugget. Each nugget must directly answer the query. The list of nuggets must not contain redundant information. Return a list of nuggets such that each nugget is on a new line. Do not number or bullet the list. Do not include anything in your response except for the list of nuggets. Here is an example of the output format:
nugget1
nugget2
...
Query: q
Context: c
LLM:
nugget1
nugget2
...

Table 8: Prompt for QGD_{all} and QRD_{all}.

SYSTEM: You are NuggetGenerateLLM, an AI assistant specialized in using context to update a list of all information nuggets that are required to completely answer a given query. A nugget is an atomic fact.
USER: Use the context provided to update the list of information nuggets, if needed. The list should contain all nuggets that are required to completely answer the query given below. If no list of nuggets is provided, generate a list of nuggets. Each nugget must contain one, and only one, fact. A nugget must be as concise and as specific as possible. A nugget cannot contain a list, each element in a list must be its own nugget. Each nugget must directly answer the query. The list of nuggets must not contain redundant information. Return a list of nuggets such that each nugget is on a new line. Do not number or bullet the list. Do not include anything in your response except for the list of nuggets. Here is an example of the output format:
nugget1
nugget2
...
Query: q
Context: c
Initial Nugget List: i
LLM:
nugget1
nugget2
...

Table 9: Prompt for QGD_{seq} and QRD_{seq}.

SYSTEM: You are NuggetGenerateLLM, an AI assistant specialized in using context to generate all information nuggets that are required to completely answer a given query. A nugget is an atomic fact.
USER: Use the context provided to generate all the information nuggets that are required to completely answer the query given below. Each nugget must contain one, and only one, fact. A nugget must be as concise and as specific as possible. A nugget cannot contain a list, each element in a list must be its own nugget. Each nugget must directly answer the query. The list of nuggets must not contain redundant information. Return a list of nuggets such that each nugget is on a new line. Do not number or bullet the list. Do not include anything in your response except for the list of nuggets. Here is an example of the output format:
nugget1
nugget2
...
Query: q
Answer: a
Context: c
LLM:
nugget1
nugget2
...

Table 10: Prompt for $QAGD_{all}$ and $QARD_{all}$.

SYSTEM: You are NuggetGenerateLLM, an AI assistant specialized in using context to update a list of all information nuggets that are required to completely answer a given query. A nugget is an atomic fact.
USER: Use the context provided to update the list of information nuggets, if needed. The list should contain all nuggets that are required to completely answer the query given below. If no list of nuggets is provided, generate a list of nuggets. Each nugget must contain one, and only one, fact. A nugget must be as concise and as specific as possible. A nugget cannot contain a list, each element in a list must be its own nugget. Each nugget must directly answer the query. The list of nuggets must not contain redundant information. Return a list of nuggets such that each nugget is on a new line. Do not number or bullet the list. Do not include anything in your response except for the list of nuggets. Here is an example of the output format:
nugget1
nugget2
...
Query: q
Answer: a
Context: c
Initial Nugget List: i
LLM:
nugget1
nugget2
...

Table 11: Prompt for $QAGD_{seq}$ and $QARD_{seq}$.

Literary Evidence Retrieval via Long-Context Language Models

Katherine Thai and Mohit Iyyer
UMass Amherst University of Maryland, College Park,
kbthai@umass.edu, miyyer@umd.edu

https://github.com/katherinethai/long_context_relic

Abstract

How well do modern long-context language models understand literary fiction? We explore this question via the task of *literary evidence retrieval*, repurposing the RELiC dataset of Thai et al. (2022) to construct a benchmark where the entire text of a primary source (e.g., *The Great Gatsby*) is provided to an LLM alongside literary criticism with a missing quotation from that work. This setting, in which the model must generate the missing quotation, mirrors the human process of literary analysis by requiring models to perform both global narrative reasoning and close textual examination. We curate a high-quality subset of 292 examples through extensive filtering and human verification. Our experiments show that recent reasoning models, such as GEMINI PRO 2.5 can exceed human expert performance (62.5% vs. 50% accuracy). In contrast, the best open-weight model achieves only 29.1% accuracy, highlighting a gap in interpretive reasoning. Despite their speed and apparent accuracy, even the strongest models struggle with nuanced literary signals and overgeneration, signaling open challenges for applying LLMs to literary analysis. We release our dataset and evaluation code to encourage future work in this direction.

1 Introduction

The emergence of long-context language models, which can process millions of tokens (Gemini Team, 2024), has unlocked new AI applications for literary analysis. In this paper, we focus on the task of *literary evidence retrieval*, in which a model must retrieve a supporting quotation from a primary source (e.g., a novel) to substantiate an excerpt of literary criticism.

Thai et al. (2022) frame literary evidence retrieval as a computational task by introducing RELiC, a dataset that contains short excerpts from published literary criticism that include quotations from famous novels. While RELiC was developed to benchmark and improve retriever models, which compute embeddings of claims and short chunks of the book text, we repurpose it as a testbed for long-context[1] language models: A model is given the entire text of the book and an excerpt of literary criticism with a missing quotation from that book, then asked to generate the missing quote (see Figure 1, top). This task does not exactly mirror the human process of literary analysis, in which a scholar typically develops claims and selects supporting quotes iteratively. However, the task requires the same understanding of and complex reasoning over plot, subtext, and other literary devices with the advantage of being easily verifiable.

To enable robust evaluation, we curated a high-quality subset of 292 examples from RELiC, verified through a combination of automated filtering and expert human review. These 292 examples contain claims that require both global reasoning over events and "close reading" over singular passages to solve. Our experiments reveal that state-of-the-art reasoning models significantly outperform previous LLMs and even surpass a human expert baseline: our best model, GEMINI PRO 2.5 obtains an accuracy of **62.5%** compared to **55.0%** for a human expert on a subset of the data. However, we also find that these models tend to overgenerate and struggle with subtle literary cues. Open-weight models perform substantially worse, suggesting that interpretive reasoning, not just long-context capacity, is essential to success in this domain, forming important directions for future research.

2 Dataset Curation

The RELiC dataset (Thai et al., 2022) consists of 78k excerpts of English-language literary analysis collected from scholarly journals where each excerpt includes a direct quotation from one of 79

[1] In this paper, we consider "long-context" to mean allowing a minimum of 128k tokens as input.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 369–380
July 27 - August 1, 2025

MODEL + SIMPLE PROMPT	ALL (n=292)	👤 (n=40)	📖 (n=39)
GEMINI PRO 2.5(Google, 2025)	63.7	57.5	**82.1**
O3(OpenAI, 2025b)	49.7	47.5	71.8
GEMINI PRO 1.5 (Gemini Team, 2024)	36.1	22.5	43.6
O1 (OpenAI, 2024)	32.2	25.0	41.0
GPT-4O (OpenAI et al., 2024)	27.4	17.5	35.9
QWEN 2.5 INSTRUCT (72B) (Qwen et al., 2025)	11.3	7.5	20.5
LLAMA 3.1 INSTRUCT (8B) (Grattafiori et al., 2024)	5.1	5.0	2.6
QWEN 2.5 INSTRUCT (7B)	2.7	5.0	2.6
LLAMA 3.3 INSTRUCT (70B)	2.1	0.0	2.6
MODEL + EXPLANATION PROMPT			
GEMINI PRO 2.5	**64.7**	**62.5**	79.5
GPT-4.1(OpenAI, 2025a)	51.0	47.5	69.2
O3	50.7	50.0	66.7
GEMINI PRO 1.5	38.5	40.0	50.0
CLAUDE SONNET 3.7(Anthropic, 2025)	37.0	32.5	48.7
DEEPSEEK-R1(DeepSeek-AI et al., 2025)	29.1	15.0	38.5
GPT-4O	24.3	22.5	31.8
QWEN 3 (32B)(Qwen et al., 2025)	19.2	20.0	33.3
QWEN 3 (8B)	8.9	5.0	10.3
O3-MINI (OpenAI, 2025c)	8.3	10.0	13.6
BASELINES			
GTE-QWEN2-7B-INSTRUCT	4.5	2.5	6.8
HUMAN	-	55.0	-

Table 1: Percentage of test set examples where the model generated the correct ground truth quotation for different folds of the test set.

	Primary Sources (n=7)		Dataset Examples (n=292)		
	TOKENS	WORDS	TOKENS	WORDS	# EX/BOOK
MEAN	85,526	69,456	254.9	203.6	36.0
ST. DEV.	26,6304	21,167	66.9	52.9	19.6
MAX	124,544	102,549	492.0	385.0	52.0
MIN	45,038	37,209	116.0	91.0	7.0

Table 2: Summary statistics for long-context RELiC. Token counts were computed with the `o200k_base` encoding via `tiktoken` (`https://github.com/openai/tiktoken`) and word counts were computed by splitting on whitespace.

primary source texts in the public domain. Each example in the dataset contains up to four sentences preceding the quotation and up to four sentences following the quotation. Together, these sentences of make up the "context" of the ground truth quotation. The primary source quotations may be up to five consecutive sentences in length.

2.1 Adapting RELiC for long-context reasoning

While large, RELiC is also noisy, which necessitates several filtering steps before we can evaluate models on it. We implemented an extensive data preprocessing pipeline[2] that involved several cleaning and filtering passes with GPT-4O-MINI and GPT-4O supplemented by some programmatic heuristics, all geared towards mitigating the below issues:

Low data quality: Some RELiC examples contain **OCR artifacts** present in the primary source texts that render the prefixes and suffixes ungrammatical. There are also some examples that are **misclassified** as literary analysis.

Model exploits: Several aspects of RELiC examples can provide unintended cues to models, allowing them to bypass the reasoning challenge. One was the **disclosure of the location** of the quote in the prefix or suffix. Another was **quote leakage**, or the appearance of part or all of the ground truth quotation in the context. Finally, **data contamination** was a concern–the literary analysis excerpts could appear in the training data of the LLMs we benchmarked and may have been memorized, or the ground truth quotation is so prevalent in the training data (because it belongs to a public domain novel) that the model is able to retrieve it without needing the primary source text at all.

Human verification: After filtering, we create a high-quality human-verified subset of the dataset by having one of the authors, who has a degree in English literature and has read all of the primary source novels manually review 400 filtered examples. This author marked 292 examples that were well-formed instances of literary analysis with a ground truth quotation that could be identified given only the book and literary analysis context. See Table 2 for dataset statistics.

Dataset folds: Our new dataset contains two labeled folds of special data, (1) the 👤 HUMAN EVAL SET: 40 examples attempted by our author, and (2) the 📖 CLOSE READING SET: 39 examples labeled by our author as examples of **close reading**, a literary analysis technique in which the reader considers "linguistic elements, semantic aspects, syntax, rhetoric, structural elements, thematic, and generic references in the text" (Ohrvik, 2024). A natural consequence of this interpretive technique is the repeated citation of parts of the ground truth quote in the context. Since these examples contain lexical overlap with the ground truth quotation, models can exploit this during evidence retrieval.

3 Experimental Setup

We evaluated four closed-source and four open-weight models on the long-context RELiC dataset, prompting each in a zero-shot setting to retrieve the most fitting quotation for a literary analysis excerpt given the full primary source text. We tested two prompt types: (1) SIMPLE, which requested only the quotation, and (2) EXPLANATION, which first asked the model to justify its choice before

[2]All prompts, details of filtering heuristics, and descriptions of our manual validations are in Appendix A.

Literary Analysis Excerpt with "Ground Truth Quotation" from *The Scarlet Letter*

Chillingworth is necessary to the tale and convincing enough, but he cannot shake off certain articles of dress and look that for Hawthorne were literary conventions. He is that unreal thing, a villain; he is Archimago still, stooping as he walks.

"A writhing horror twisted itself across his features, like a snake gliding swiftly over them, and making one little pause, with all its wreathed intervolutions in open sight. His face darkened with some powerful emotion, which, nevertheless, he so instantaneously controlled by an effort of his will, that, save at a single moment, its expression might have passed for calmness. After a brief space, the convulsion grew almost imperceptible, and finally subsided into the depths of his nature."

This was when he stood by the scaffold and recognized Hester Prynne. He had come there out of melodrama, and in melodrama he remains. If Hawthorne makes him acceptable nevertheless, the reason is the vitality of the whole world here envisioned.

✅ **Human** explanation:

I was between this and the description of Chillingworth's shoulders being uneven, but I think this supports the "melodrama" mentioned in the suffix better than the description of his uneven shoulders. It contains a very exaggerated description of Chillingworth; words that connote movement like "writhing," "twisted," "gliding swiftly," and "convulsion"; and a tension between "powerful emotion" and a struggle for "calmness" that all contribute to the melodrama of the scene.

❌ **`Gemini 2.5 Pro`, `o3`, and `GPT-4.1`** response:

He was small in stature, with a furrowed visage, which, as yet, could hardly be termed aged. There was a remarkable intelligence in his features, as of a person who had so cultivated his mental part that it could not fail to mould the physical to itself, and become manifest by unmistakable tokens. Although, by a seemingly careless arrangement of his heterogeneous garb, he had endeavored to conceal or abate the peculiarity, it was sufficiently evident to Hester Prynne, that one of this man's shoulders rose higher than the other.

Figure 1: An example from long context RELiC where the HUMAN annotator successfully retrieves the quotation while the top-performing LLMs fail.

selecting a quotation[3]. Our prompts are adapted from Karpinska et al. (2024) in the NOCHA long-context benchmark. Additionally, we implemented an embedding-based retrieval baseline using GTE-QWEN2-7B-INSTRUCT, the top-performing text embedding model on MTEB[4] at the time of writing.

3.1 Human Evaluation

The author who manually validated high quality examples also attempted 40 examples from four previously-read primary source novels. The author had access to a digital copy of the primary source text with a string-matach search function and the corresponding Wikipedia summaries. In addition to selecting missing quotations, the author also wrote a short justification for each choice. The task took approximately 8 hours.

3.2 Evaluation Scheme

Our automatic evaluation of model responses applied partial ratio fuzzy matching (to account for minor typographical differences) to check that the model response was from the primary source text and to measure overlap between the response and the ground truth quotation[5]. For the embedding baseline, we calculate recall@1.

[3]Prompts & inference details in Appendix C.

[4]`https://huggingface.co/spaces/mteb/leaderboard` accessed February 2025

[5]Further details can be found in Appendix D.

4 Results & Analysis

Table 1 reports results for all evaluated models on the full set of 292 claims and the two folds.

LLMs outperform HUMAN and SOTA embedding baseline Google's GEMINI PRO 2.5 reasoning model achieves better accuracy than our HUMAN baseline in a fraction of the time—each API call took an average of 45 seconds vs. an average of 12 minutes for the HUMAN expert. Both Google and OpenAI's latest long-context models show substantial performance gains over their immediate predecessors on the literary evidence retrieval task. GEMINI PRO 2.5 outperforms GEMINI PRO 1.5 by a wide margin (64.7% vs. 38.5%), and OpenAI's GPT-4.1 improves significantly over GPT-4O (51.0% vs. 24.3%). Similarly, O3 surpasses O1 (48.7% vs. 32.2%). The embedding-based method achieved 4.5% accuracy, only 1.6% higher than the best recall@1 reported in the original RELiC paper from three years ago. LLMs' success over embeddings suggests the importance of contextualized representations that can enable reasoning over entire primary source texts.

Closed-source outperform open-weight LLMs Despite recent progress in open-weight LLM development, we observe a striking performance gap between closed-source and open-weight models on the literary evidence retrieval task. The best-performing open-weight model, DEEPSEEK-R1,

Literary Analysis Excerpt with "Ground Truth Quotation" from *Frankenstein*

"It may therefore be judged indecent in me to come forward on this occasion," she proclaims on the stand, "but when I see a fellow creature about to perish through the cowardice of her pretended friends, I wish to be allowed to speak, that I may say what I know of her character." Attempting to resist class lines and to speak of Justine as a "fellow creature" calls the silent cowardly, which means that Elizabeth, too, forgot her manners. Her testimony of Justine's dedication, loyalty, and nurturing cannot be heard, and the jury's response mimics Frankenstein's own preoccupation and inability to read the interactions and relationships between women in the broader sense.

"But I did not feel the inconvenience of the weather; my imagination was busy in scenes of evil and despair."

Class and gender clearly matter in Geneva, and these factors, rather than a psychosexual rivalry, explain why Justine is punished for her ingratitude toward her benefactors. Elizabeth's testimony falls on deaf ears. What the jury perceives as important, and punishes, is indecorum—a breach of the mistress-servant relationship.

✗ **Human** and **Gemini 2.5 Pro** response:

A murmur of approbation followed Elizabeth's simple and powerful appeal, but it was excited by her generous interference, and not in favour of poor Justine, on whom the public indignation was turned with renewed violence, charging her with the blackest ingratitude.

✗ **Human** explanation:

I selected the quote that follows Elizabeth's full testimony, which is quoted in part at the beginning of the prefix. The selected quote also mentions Justine's "ingratitude," like the suffix.

✗ **Gemini 2.5 Pro** explanation:

The literary analysis argues that Elizabeth's testimony for Justine, despite its appeal to see Justine as a "fellow creature," fails because the jury (and public) focuses on Justine's perceived "ingratitude" and breach of class decorum rather than her actual character. The chosen window directly follows Elizabeth's speech in court and describes the public's reaction: while Elizabeth's act of speaking earns some approval for its "generous interference," it does not sway opinion in Justine's favor.

Figure 2: An example of long context RELiC where the HUMAN annotator and GEMINI PRO 2.5 both select the same quotation, but it is not the ground truth quotation.

achieves only 29.1% accuracy—less than half the performance of the top closed-source model, GEMINI PRO 2.5 (64.7%).

Small LLMs can't capitalize on close reading examples Nearly all models larger than 8B parameters show modest to significant improvements on the close reading fold, with some models improving by over 20%. This effect is expected due to ground truth leakage in the the provided excerpt of literary analysis, but the 7B and 8B parameter open-weight models experience nearly no performance boost on this fold—the performance of LLAMA 3.1 INSTRUCT (8B) actually suffers, suggesting that the smaller models lack the capacity to even exploit lexical overlap.

EXPLANATION prompt provides a glimpse into model reasoning We observed mixed results when querying non-reasoning LLMs with the EXPLANATION prompt. While GEMINI PRO 1.5 improved with explanation-based reasoning, GPT-4O saw a decline. Recent reasoning LLMs such as GEMINI PRO 2.5 and O3 use internal reasoning tokens that are not exposed via the API but influence the model's final output. These tokens are critical to the model's performance on complex tasks, but their inaccessibility complicates model evaluation and comparison. Applying the EXPLANATION prompt to these reasoning models preserved accuracy, despite the redundancy, while providing insight into the models' reasoning processes. See a comparison of human and model justifications in Figure 2.

All models tend to overgenerate Despite prompts explicitly instructing models to select and generate no more than five consecutive sentences from the primary source, we observe a consistent tendency across all evaluated models to produce significantly longer outputs. To quantify this overgeneration behavior, we compute the length ratio, defined as the ratio of the model's generated output length to the ground truth quotation length (measured in characters); values closer to 1.0 indicate greater adherence to the prompt constraints and minimal overgeneration. The length ratios are reported in Table 3.

We find that all models overgenerate, including the human annotator (average ratio of 2.1), likely due to natural variance in interpreting sentence boundaries. However, language models consistently surpass this baseline: for instance, state-of-the-art models such as GEMINI PRO 2.5 and GPT-4.1 have average ratios of 3.0 and 4.8, respectively. Smaller open models like LLAMA 3.1 INSTRUCT (8B) and LLAMA 3.3 INSTRUCT (70B) exhibit even more extreme overgeneration (ratios > 5.7), suggesting that weaker models may compensate for uncertainty by producing longer outputs.

LLMs struggle with literary nuance In Figure 1, we present a challenging example where all LLMs fail but our HUMAN expert correctly identified the ground truth quotation from *The Scarlet Letter*, a >80k-word novel. The context alludes to a description of the character Roger Chillingworth at a specific moment in the story (when he recognizes Hester Prynne). In their explanation, the HUMAN expert mentions considering two different passages, but ultimately selecting the correct passage because it best demonstrates the "melodrama" mentioned in the literary analysis. The expert highlights the features of the correct passage that demonstrate "melodrama": an exaggerated character description, words that connote movement, and emotional tension. All three top-performing LLMs selected the other passage the HUMAN expert was considering but ultimately eliminated, suggesting that even the best models still lack the expertise to navigate literary devices and more nuanced signals that inform expert literary interpretation.

Models can identify alternative literary evidence In Figure 2, we present an example where both HUMAN and GEMINI PRO 2.5 selected the same "incorrect" quote. The explanations reveal that both interpreted the literary analysis context as an introduction to a quote about the jury's response. However, the ground truth quotation actually supports Frankenstein's own preoccupation rather than the trial's immediate outcome. This example highlights two key insights: (1) literary evidence retrieval is inherently interpretative, meaning that multiple passages may plausibly support a given claim, and (2) LLMs, like human readers, can surface alternative yet reasonable quotations that align with certain aspects of the analysis. While models may not always select the canonical answer, their ability to propose viable alternative evidence could be valuable for assisting literary scholars in exploring multiple textual connections.

5 Related Work

Our work builds on recent papers that apply and evaluate LLMs for computational literary analysis. Prior work has explored summarization or claim verification in novels (Subbiah et al., 2024a; Kim et al., 2024; Karpinska et al., 2024) or short stories (Subbiah et al., 2024b). Other more specific tasks, mainly in the short context setting, include extracting narrative elements (Shen et al., 2024), story arcs and turning points (Tian et al., 2024), character analysis (Papoudakis et al., 2024), narrative discourse (Piper and Bagga, 2024), plot development (Huot et al., 2024; Xu et al., 2024), and creativity evaluation (Chakrabarty et al., 2024).

MODEL + SIMPLE PROMPT	Accuracy (n=40)	Avg. Length Ratio
GEMINI PRO 2.5	57.5	3.5
O3	47.5	3.5
GEMINI PRO 1.5	22.5	2.8
O1	25.0	3.2
GPT-4O	17.5	3.9
QWEN 2.5 INSTRUCT (72B)	7.5	3.2
LLAMA 3.1 INSTRUCT (8B)	5.0	5.9
QWEN 2.5 INSTRUCT (7B)	5.0	2.8
LLAMA 3.3 INSTRUCT (70B)	0.0	5.7
MODEL + EXPLANATION PROMPT		
GEMINI PRO 2.5	**62.5**	3.0
GPT-4.1	47.5	4.8
O3	50.0	2.7
GEMINI PRO 1.5	40.0	3.3
CLAUDE SONNET 3.7	32.5	4.0
DEEPSEEK-R1	15.0	3.6
GPT-4O	22.5	3.6
QWEN 3 (32B)	20.0	2.7
QWEN 3 (8B)	5.0	2.4
O3-MINI	10.0	3.5
BASELINES		
HUMAN	55.0	**2.1**

Table 3: The average length ratios for each model, defined as the ratio of the length model generation to that of the ground truth (measured in characters).

6 Conclusion

In this work, we introduced a long-context dataset derived from RELiC for evaluating literary evidence retrieval using large language models. We constructed a high-quality test set of 292 examples, ensuring rigorous evaluation through filtering and human verification. Our results demonstrate that long-context LLMs outperform embedding-based retrieval methods and even a HUMAN expert baseline. However, closer analysis reveals that model success is not always grounded in robust interpretative abilities: models tend to overgenerate and still struggle to grasp nuances in literature. However, they can also successfully identify additional evidence for literary claims. Finally, open-weight models still lag far behind their closed-weight counterparts, suggesting that long-context capabilities alone are insufficient without strong interpretive reasoning. We release our data and evaluation framework to spur research at the intersection of NLP and literary analysis.

Limitations

As noted in the original paper, the RELiC dataset represents a limited, English-only subset of world literature, with a strong emphasis on the Western

literary canon. To foster a more inclusive and representative benchmark, we aim to extend this work to cross-lingual literary analysis and retrieval on texts from historically underrepresented literary traditions. Expanding the dataset in this way would not only enhance the linguistic and cultural diversity of literary evidence retrieval but also improve the generalizability of models across different literary frameworks.

Additionally, our human evaluation was limited in scope and scale, as it was conducted by a single annotator, one of the authors of this paper. While this approach was practical given the intensive nature of literary evidence retrieval over full novels, it inherently reflects the knowledge, biases, and interpretive lens of a single individual. In the future, we aim to broaden our evaluation by incorporating multiple expert annotators, enabling a more comprehensive and diverse assessment of model performance on the task of complex literary reasoning.

Ethical Considerations

While LLMs can assist in literary interpretation, they should not replace scholarly analysis conducted by trained experts. We acknowledge the potential for misuse of a system that can retrieve evidence for scholarly claims and emphasize that automated retrieval should be viewed as a complement to, not a substitute for, human literary scholarship.

As mentioned in the Limitations section, the primary source texts in our dataset are drawn from public domain works, which primarily reflect Western literary traditions. The results of this benchmark may not generalize to literary traditions outside the Anglophone canon, potentially reinforcing existing biases in computational literary studies.

References

Anthropic. 2025. Claude 3.7 sonnet.

Tuhin Chakrabarty, Philippe Laban, Divyansh Agarwal, Smaranda Muresan, and Chien-Sheng Wu. 2024. Art or artifice? large language models and the false promise of creativity. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA. Association for Computing Machinery.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Google. 2025. Gemini 2.5 pro. Available at https://deepmind.google/models/gemini/pro/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Fantine Huot, Reinald Kim Amplayo, Jennimaria Palomaki, Alice Shoshana Jakobovits, Elizabeth Clark, and Mirella Lapata. 2024. Agents' room: Narrative generation through multi-step collaboration. *arXiv preprint arXiv:2410.02603*.

Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. One thousand and one pairs: A "novel" challenge for long-context language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17048–17085, Miami, Florida, USA. Association for Computational Linguistics.

Yekyung Kim, Yapei Chang, Marzena Karpinska, Aparna Garimella, Varun Manjunatha, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. Fables: Evaluating faithfulness and content selection in book-length summarization. In *Conference on Language Modeling*.

Ane Ohrvik. 2024. What is close reading? an exploration of a methodology. *Rethinking History*, 28(2):238–260.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2024. O1 model card. Accessed: 2024-02-16.

OpenAI. 2025a. Gpt-4.1.

OpenAI. 2025b. o3.

OpenAI. 2025c. Openai o3-mini system card. Accessed: 2025-02-15.

Argyrios Papoudakis, Mirella Lapata, and Frank Keller. 2024. BookWorm: A dataset for character description and analysis. In *Findings of the Association*

for Computational Linguistics: EMNLP 2024, pages 4471–4500, Miami, Florida, USA. Association for Computational Linguistics.

Andrew Piper and Sunyam Bagga. 2024. Using large language models for understanding narrative discourse. In *Proceedings of the The 6th Workshop on Narrative Understanding*, pages 37–46, Miami, Florida, USA. Association for Computational Linguistics.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. Qwen2.5 Technical Report. *arXiv preprint*. ArXiv:2412.15115 [cs].

Jocelyn Shen, Joel Mire, Hae Won Park, Cynthia Breazeal, and Maarten Sap. 2024. HEART-felt narratives: Tracing empathy and narrative style in personal stories with LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1026–1046, Miami, Florida, USA. Association for Computational Linguistics.

Melanie Subbiah, Faisal Ladhak, Akankshya Mishra, Griffin Thomas Adams, Lydia Chilton, and Kathleen McKeown. 2024a. STORYSUMM: Evaluating faithfulness in story summarization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9988–10005, Miami, Florida, USA. Association for Computational Linguistics.

Melanie Subbiah, Sean Zhang, Lydia B. Chilton, and Kathleen McKeown. 2024b. Reading subtext: Evaluating large language models on short story summarization with writers. *Transactions of the Association for Computational Linguistics*, 12:1290–1310.

Katherine Thai, Yapei Chang, Kalpesh Krishna, and Mohit Iyyer. 2022. RELiC: Retrieving evidence for literary claims. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7500–7518, Dublin, Ireland. Association for Computational Linguistics.

Yufei Tian, Tenghao Huang, Miri Liu, Derek Jiang, Alexander Spangher, Muhao Chen, Jonathan May, and Nanyun Peng. 2024. Are large language models capable of generating human-level narratives? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17659–17681, Miami, Florida, USA. Association for Computational Linguistics.

Liyan Xu, Jiangnan Li, Mo Yu, and Jie Zhou. 2024. Fine-grained modeling of narrative context: A coherence perspective via retrospective questions. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5822–5838, Bangkok, Thailand. Association for Computational Linguistics.

A Dataset Details

We are using a publicly available dataset released by researchers who stated in their paper that they released their data to "facilitate further research in this direction." Statistics for the primary sources can be found in 4.

B LLM-Aided Data Processing

B.1 Filtering steps

These cleaning and filtering steps were applied to the entire RELiC dataset in the following order:

1. `CLEAN` (GPT-4O-MINI): We asked the model to remove any OCR artifacts and to ensure the context and ground truth quotation flow seamless and grammatically without changing any meaning. Additionally, we asked the model to remove any remaining in-line citations that revealed the page number of the ground truth quotation, which allowed us to preserve some examples that might have been caught by the `LOCATION` filter below. See the prompt in Table 8.

2. `LEAKAGE` (heuristic): If the sentences in the context preceding or following the ground truth quotation were fuzzy-matches (threshold: 95) with any text from the primary source, the example was excluded from our dataset.

3. `LIT ANALYSIS` (GPT-4O-MINI): We asked the model to classify whether each RELiC instance was an example of literary analysis. If it was not, we excluded it from our dataset. See the prompt in Table 9.

4. `LOCATION` (GPT-4O-MINI): We asked the model if the context revealed the location of the ground truth quotation. If it did, we excluded it from our dataset. See the prompt in Table 10.

5. `FIRST SENT` (heuristic): We identified cases where the ground truth quotation was the first sentence of its primary source novel. The intuition behind this filter (and the following two) was that the first sentence of a primary source might be famous or more likely to be quoted, and therefore easier to guess without needing to reason over the text or more likely to appear in training data.

Book Title	Author	Publication Year	Token Count	Word Count
Brave New World	Aldous Huxley	1932	91,472	65,278
What Maisie Knew	Henry James	1897	124,544	95,988
Ethan Frome	Edith Wharton	1911	45,038	34,926
Frankenstein	Mary Shelley	1818	95,018	75,131
The Great Gatsby	F. Scott Fitzgerald	1925	63,683	48,972
The Awakening	Kate Chopin	1899	66,574	49,932
The Scarlet Letter	Nathaniel Hawthorne	1850	112,355	83,311

Table 4: Books included in the dataset. The token count was provided as per `tiktoken` tokenization.

6. `LAST SENT` (heuristic): Similarly, we identified cases where the ground truth quotation was the last sentence of its primary source novel.

7. `OUTLIER` (heuristic): We identified cases where the ground truth quotation was cited much more frequently than any of the other quotations from the same primary source novel.

8. `EZ2MEM` (GPT-4): We asked the model to perform the RELiC task *without* providing the primary source text and identified cases where the model was able to correctly generate at least one sentence of the ground truth quotation.

Note that we did not automatically exclude any of the examples identified by the last four filters. Though we did not use them in this project, we have left the labels in our dataset to facilitate future research.

B.2 Manual validation of data

We conducted multiple validations of the RELiC data for quality control using human annotations.

Validation of LLM filters To validate our LLM-aided approach to data preprocessing, we manually annotated 100 examples of RELiC and compared our judgments to keep or reject each example to the results of the LLM + heuristic filters. The f1-score of our filtering scheme was 89.8, with our scheme identifying 57 true positives, 30 true negatives, 6 false positives, and 7 false negatives.

C Model Inference Details

API access details

- All OpenAI models were accessed via the OpenAI API (https://platform.openai.com/).
- All Gemini models were accessed via the Google AI API (https://ai.google.dev/).
- CLAUDE SONNET 3.7 was accessed via the Vertex AI/Google Cloud (https://cloud.google.com/vertex-ai?hl=en).
- DeepSeek was run via OpenRouter (https://openrouter.ai/).

The approximate total cost of running all closed-weight models for both development and evaluation was $2k.

Local inference details All Llama and Qwen models were run locally. The smaller open-weight models were run on 1 A100 GPU for a total of 14 hours, while the larger open-weight models were run on 4 A100 GPUs for a total of 16 hours.

Generation hyperparameters The OpenAI reasoning models (O1, O3-MINI, and O3) were a special case: the temperature parameter is fixed and the token generation limit includes the inaccessible reasoning tokens. As such, we set the token generation limit to 12,000 for O1 and O3-MINI. For O3, the token generation limit was set to 25,000 as recommended by OpenAI[6]. For all three OpenAI reasoning models, we used the default `medium` reasoning effort.

For all other models, the token generation limit was set to 800 for SIMPLE prompts and 1,200 for EXPLANATION prompts. Temperature was set to 0.0 for all models except for QWEN 3 (32B) and QWEN 3 (8B), where temperature was set to the default 0.6.

For a list of the models and their checkpoints, see Table 5.

D Evaluation Scheme Details

We use the `rapidfuzz` Python package for our fuzzy match evaluation with a threshold of 95 for

[6] https://platform.openai.com/docs/guides/reasoning#allocating-space-for-reasoning

Model	Context	Avail.	Checkpoints
o1	200k	🔒	o1-2024-12-17
o3	200k	🔒	o3-2025-04-16
o3-mini	200k	🔒	o3-mini-2025-01-31
GPT-4o	128k	🔒	gpt-4o-2024-11-20
GPT-4.1	1M	🔒	gpt-4.1-2025-01-31
Gemini Pro 1.5	1M	🔒	gemini-1.5-pro-002
Gemini Pro 2.5	1M	🔒	gemini-2.5-pro-preview-05-06
Claude Sonnet 3.7	200k	🔒	claude-3-7-sonnet@20250219
Llama 3.3 Instruct (70B)	128k	🔓	Llama-3.3-70B-Instruct
Llama 3.1 Instruct (8B)	128k	🔓	Llama-3.1-8B-Instruct
Qwen 2.5 Instruct (72B)	128k	🔓	Qwen2.5-72B-Instruct
Qwen 2.5 Instruct (7B)	128k	🔓	Qwen2.5-7B-Instruct
Qwen 3 (8B)	131k*	🔓	Qwen3-8B
Qwen 3 (32B)	131k*	🔓	Qwen3-32B
gte-Qwen2-7B-instruct (7B)	32k	🔓	gte-Qwen2-7B-instruct

Table 5: The upper rows display the evaluated LLMs, while the bottom row displays the text embedding model used for the baseline. Context lengths marked with * were extended with YaRN.

checking the existence of the model response in the primary source and a threshold of 90 for checking the overlap between the model response and the ground truth quotation. The fuzzy matching allows for small typographical differences between the primary source and the model outputs (e.g. different types of quotation marks). Thresholds were determined after manual inspection of outputs at varying thresholds.

E Use of AI Assistants

The authors used Github Copilot for coding assistance during their experiments and ChatGPT for assistance in formatting LaTeX in the writing of this paper.

Prompt (Simple)
You are provided with the full text of **[book_title]** and an excerpt of literary analysis that directly cites **[book_title]** with the cited quotation represented as <MASK>. Your task is to carefully read the text of **[book_title]** and the excerpt of literary analysis, then select a window from **[book_title]** that most appropriately replaces <MASK>as the cited quotation by providing textual evidence for any claims in the literary analysis. The excerpt of literary analysis should form a valid argument when <MASK>is replaced by the window from **[book_title]**. <full_text_of_[book_title_snake_case]>**[book_sentences]** </full_text_of_[book_title_snake_case]> <literary_analysis_excerpt>**[lit_analysis_excerpt]** </literary_analysis_excerpt> Identify the window that best supports the claims being made in the excerpt of literary analysis. The window should contain no more than 5 consecutive sentences from **[book_title]**. Provide your final answer in the following format: <window>**YOUR SELECTED WINDOW** </window>

Table 6: Prompt template for literary evidence retrieval (Simple).

Prompt w/ Explanations
You are provided with the full text of **[book_title]** and an excerpt of literary analysis that directly cites **[book_title]** with the cited quotation represented as <MASK>. Your task is to carefully read the text of **[book_title]** and the excerpt of literary analysis, then select a window from **[book_title]** that most appropriately replaces <MASK>as the cited quotation by providing textual evidence for any claims in the literary analysis. The excerpt of literary analysis should form a valid argument when <MASK>is replaced by the window from **[book_title]**. <full_text_of_[book_title_snake_case]>**[book_sentences]** </full_text_of_[book_title_snake_case]> <literary_analysis_excerpt>**[lit_analysis_excerpt]** </literary_analysis_excerpt> First, provide an explanation of your decision marking process in no more than one paragraph. Then, identify the window that best supports the claims being made in the excerpt of literary analysis. The window should contain no more than 5 consecutive sentences from **[book_title]**. Provide your final answer in the following format: <explanation>**YOUR EXPLANATION** </explanation> <window>**YOUR SELECTED WINDOW** </window>

Table 7: Prompt template for literary evidence retrieval with explanation.

Prompt for Cleaning RELiC Examples
I want to create a dataset for Natural Language Processing that consists of excerpts of literary analysis that quote directly from a primary source text. I have already collected windows of literary analysis that quote from many different primary sources in the public domain. The part of the excerpt before the quotation is called the "prefix," and the part of the excerpt after the quotation is called the "suffix." Because I collected the data from PDFs with OCR, the prefixes and suffixes contain artifacts like in-line citations, page numbers, chapter names, headers, and footers that I need to remove. Here is an example: <prefix>**[prefix]** </prefix> <ground_truth_quotation>**[ground_truth_quotation]** </ground_truth_quotation> <suffix>**[suffix]** </suffix> Please follow the following guidelines to help me clean and filter this example: 1. Remove all textual artifacts from the OCR process by deleting them. Artifacts include page numbers, chapter headings, footnotes, image captions, etc. 2. Correct the grammar, punctuation, and spelling of the prefix and suffix without altering the meaning so that the primary source quotation fits seamlessly and grammatically between the prefix and suffix. 3. Remove all in-line citations following quotations, especially those that say things like "(my emphasis)." Respond with ONLY the cleaned prefix and suffix in the following format: <clean_prefix>**CLEAN PREFIX** </clean_prefix> <clean_suffix>**CLEAN SUFFIX** </clean_suffix>

Table 8: Prompt template for cleaning RELiC examples

Prompt for Classifying Literary Analysis
I want to create a dataset for Natural Language Processing that consists of excerpts of literary analysis that quote directly from a primary source text. Other texts can contain quotes from primary sources, like biographies of authors. However, I only want examples of literary analysis in the dataset. Please determine whether the following excerpt is an example of literary analysis. If it is literary analysis, respond with TRUE. Otherwise, respond with FALSE. Here is the excerpt: <excerpt>**[clean_prefix] [answer_quote] [clean_suffix]** </excerpt> Respond with ONLY your answer in the following format: <answer>**YOUR ANSWER** </answer>

Table 9: Prompt template for classifying literary analysis

Prompt for Identifying Quote Location Disclosure

I want to create a dataset for a Natural Language Processing task called "Literary Evidence Retrieval."

The input has two parts:

(1) An excerpt of literary analysis that quotes 1 to 5 full sentences from a primary source text, but the quote is replaced with "[MASK]".

(2) The full text of the primary source.

The output should be the correct quotation of 1 to 5 full sentences from the primary source—this is the ground truth.

I have already collected windows of literary analysis that quote from many different primary sources in the public domain.

The part of the excerpt before the quotation is called the "prefix," and the part after is called the "suffix."

To avoid giving the model hints, it's extremely important that the prefix and suffix do not mention the chapter where the ground truth quotation is located.

Please determine whether the following instance contains the chapter location of the ground truth quotation in either the prefix or the suffix. If it does, respond with TRUE; otherwise, respond with FALSE.

The prefix and suffix may contain the location of other quotations in the passage. I only want to identify if the location of the ground truth quotation is revealed.

Look out for phrases like "In the prologue" or "In Chapter..."

Here is the instance:

<prefix>**[prefix]** </prefix>

<ground_truth_quotation>**[ground_truth_quotation]** </ground_truth_quotation>

<suffix>**[suffix]** </suffix>

Respond with ONLY your answer in the following format:

<answer>**YOUR ANSWER** </answer>

Table 10: Prompt template for identifying quote location disclosure

A Little Human Data Goes A Long Way

Dhananjay Ashok[†], **Jonathan May**[†]
[†]Information Sciences Institute, University of Southern California
{ashokd, jonmay}@isi.edu

Abstract

Faced with an expensive human annotation process, creators of NLP systems increasingly turn to synthetic data generation. While this method shows promise, the extent to which synthetic data can replace human annotation is poorly understood. We investigate the use of synthetic data in Fact Verification (FV) and Evidence-based Question Answering (QA) by incrementally replacing human-generated data with synthetic points on eight diverse datasets. Strikingly, replacing up to 90% of the training data only marginally decreases performance, but replacing the final 10% leads to severe declines. We find that models trained on purely synthetic data can be improved by including as few as **125** human generated data points. We show that matching the performance gain of a little human data requires an order of magnitude more synthetic data, and then estimate price ratios at which human annotation would be a more cost-effective solution. Our results suggest that even when human annotation at scale is infeasible, there is great value to having a small proportion of the dataset being human-generated.

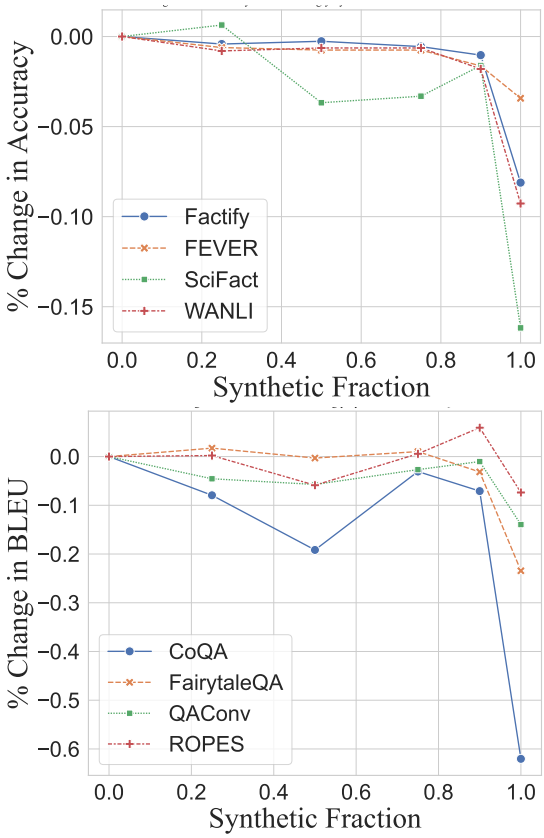


Figure 1: Change in model performance as the proportion of synthetic points in the training data is increased. Across datasets, the performance decrease when moving the synthetic proportion from 0 to 0.90 is often less than that of moving from 0.9 to purely synthetic data.

1 Introduction

From BERT (Devlin et al., 2019) to GPT-4 (Achiam et al., 2023), the explosive growth of language models (LMs) has been underpinned by exponential increases in the size of available training data. However, the more complex and specialized the task, the more expensive and challenging it is to collect human-generated data at scale (Wang et al., 2021). Combined with growing concerns that LMs may soon exhaust the stock of publicly available training data (Villalobos et al., 2024), many turn to synthetic data generation, hoping to eliminate their reliance on human annotation.

Synthetic data generation has long been used to increase the amount of training data available (Simard et al., 2002; Krizhevsky et al., 2012). Early NLP approaches use rule based methods (De Gispert et al., 2005; Chen et al., 2012), paraphrasing (Wang and Yang, 2015; Kobayashi, 2018), noising (Xie et al., 2017; Wang et al., 2018), and backtranslation (Sennrich et al., 2016; Yu et al., 2018), but are limited in their capability.

Modern LMs demonstrate the capability to solve myriad NLP tasks with minimal task specific data (Brown et al., 2020; Wei et al., 2022), making them more powerful synthetic data generators. Leveraging this, synthetic data approaches have

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 381–413
July 27 - August 1, 2025

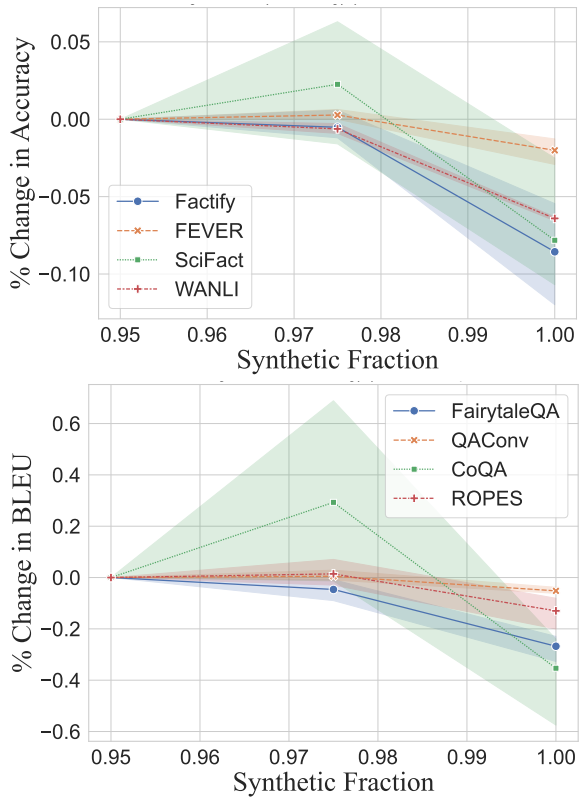


Figure 2: Model performance as the synthetic proportion of the training data varies from 0.95 to 1. Having **just 2.5%** of the training dataset being human-generated boosts performance.

seen increased use in tasks (Tan et al., 2024) such as QA (Wu et al., 2022), natural language inference (NLI) (Meng et al., 2022), text classification (Ye et al., 2022), instruction tuning (Li et al., 2024), evaluation (Dubois et al., 2024), and more (Tang et al., 2023).

The adoption has been particularly enthusiastic for tasks that require the model to 'understand' knowledge contained in an 'evidence text' e.g., FV (Tang et al., 2024), factual error correction (Ashok et al., 2023), NLI (Hosseini et al., 2024) and evidence-based QA (Schimanski et al., 2024). Such tasks are of vital importance in fake news detection (Sharma et al., 2023), retrieval-augmented generation (Gao et al., 2023) and dialogue systems (Weston et al., 2015). Recent datasets (Wu et al., 2022) and methods (Ye et al., 2022) exploit plentiful evidence texts (scientific journals, news articles, books, etc.), using synthetic generation to avoid being bottlenecked by the expensive annotation procedure (Liu et al., 2022).

Varying results across ML tasks suggest that whether completely replacing humans with synthetic data shows promise (Fan et al., 2024; Hammoud et al., 2024) or leads to failures (Bisbee et al., 2024; Guo et al., 2024) is task dependent (Li et al., 2023). In this work, we focus on **FV** and **Evidence-based QA**, performing the first investigation into the trade-offs presented by the use of synthetic data generation in these fundamental tasks.

We study eight diverse FV and QA datasets, using their 'evidence texts' to generate synthetic datasets. By holding the number of data points constant but increasing the percentage of the training data that is synthetic, we can compare the utility of synthetic data to the original human generated data points. Across multiple models, prompt models, and prompting strategies, we find (Figure 1) that while increasing the proportion of synthetic data typically causes only minor degradations in model performance, a significant decline occurs at the extremes; i.e., when the percentage of synthetic data exceeds 90%. Focusing on the extremes, we show that purely synthetically trained FV and QA systems can be meaningfully improved by including as few as **125** human-generated datapoints.

Our observations have actionable implications for researchers hoping to use synthetic data for FV and QA. The results (Figure 2, Figure 4) suggest that even when human annotation at scale is infeasible, there is great value to having a small proportion of the dataset being generated by humans.

To help guide this choice, we quantify the performance-cost tradeoff between human and synthetic data. We find (Figure 4) that matching the performance gain of just a little additional human data (only 200 data points) requires an order of magnitude more synthetic data points, empirically showing the per-data point price ratio at which human annotation is the more cost-effective solution. Finally, we conduct an analysis on the differing properties of synthetic vs. human data. Among other findings, we see that synthetic generations are longer and more extractive from the evidence texts than their human-produced counterparts.

2 Synthetic Data Generation from Evidence Texts

We study a synthetic generation pipeline representative of the methods used in the FV (Ni et al., 2024; He et al., 2023) and QA (Schimanski et al., 2024; Wan et al., 2024) literature. Using Few-Shot In-Context Learning (Brown et al., 2020), we generate synthetic (claim, label) pairs from an input evidence text. The prompt model is given exam-

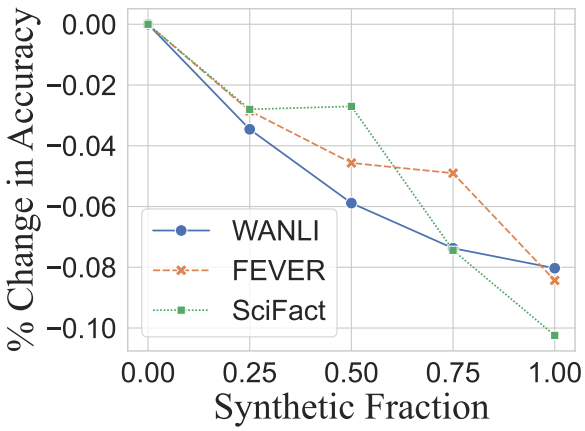


Figure 3: Change in accuracy when the test set (shown in key) is not seen during training, and the training set is a mixture of other FV datasets. Increasing the synthetic proportion of the dataset leads to performance declines even in the OOD setting, showing that human data offers genuine performance increases.

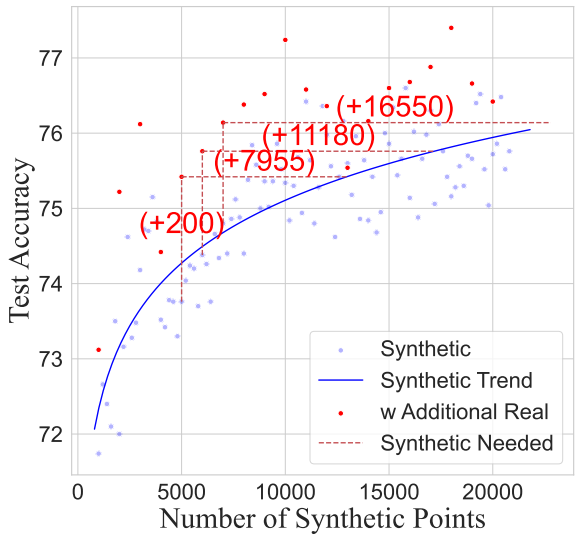


Figure 4: On the WANLI dataset, adding 200 real data points is as effective as adding an order of magnitude more synthetic data points.

ples of (evidence text, claim, label) from the human training data, and is then queried with the evidence text we seek to generate data for. QA synthetic data is generated analogously, see details in Appendix B.

In total, we use four FV/NLI datasets: FEVER (Thorne et al., 2018), SciFact (Wadden et al., 2020), WANLI (Liu et al., 2022) and FACTIFY1.0 (Mishra et al., 2022), as well as four QA datasets: ROPES (Lin et al., 2019), CoQA (Reddy et al., 2019), QAConv (Wu et al., 2022) and FairyTaleQA (Xu et al., 2022). Together, the datasets span a variety of domains (science, news, social media, reasoning, conversation, fiction). We confirm that the generations are of high quality by verifying that the diversity of the synthetic data is comparable to the human generated samples. For more details, including a discussion on data leakage, see Appendix C.

FV performance is measured by test accuracy, while QA is measured using BLEU (Papineni et al., 2002); we show robustness to choice of metric in Appendix A. Evaluation is always conducted on the (human-generated) test split of each dataset.

We use GPT-3.5-Turbo (Brown et al., 2020) for prompting and LoRA (Hu et al., 2022) on Llama3-8B (Dubey et al., 2024) for fine-tuning. Implementation details are provided in Appendix E, and our code is publicly available [1]

[1] https://github.com/DhananjayAshok/LittleHumanData/

3 Can Synthetic Data Replace Humans?

We investigate the potential of synthetic data to replace human annotation by holding the number of training data points fixed, incrementally increasing the proportion of the data that is synthetic, and fine-tuning a model on each training set.

Results: Across all datasets, using purely synthetic data leads to worse performance than the same amount of human data (Figure 1). We consider the possibility that this result could be caused by a spurious correlation between the human training and testing splits (e.g., annotation artifacts that are correlated with the label but not fundamental to the task). We conduct an out-of-distribution experiment, using different datasets for training and testing (e.g., training on FEVER + SciFACT and testing on WANLI). Increasing the synthetic proportion leads to performance declines even in the OOD setting (Figure 3), showing that human data offers genuine performance increases, and the results cannot be explained by a spurious correlation between the human test and human training samples (further discussion in Appendix A).

The performance decline is not uniform as we increase the synthetic proportion. On almost all datasets, there is only a minor degradation up until 90% replacement, after which the performance drops considerably. We zoom in on the 90%-100% interval, fixing the amount of training data at $n = 5000$ (500 for SciFact) and training on datasets with 95%, 97.5% and 100% synthetic

data (Figure 2). Surprisingly, the results show that there is a significant difference between the performance of models on 97.5% and 100% synthetic data; the addition of just **125** (2.5% of 5000) human generated datapoints reliably improves the performance of synthetically trained FV and QA models. These trends hold robustly over different languages (Arabic, Georgian, Indonesian), choice of fine-tuning model (Mistral, MPT), prompt model (GPT4 and Claude-3.5-Sonnet), prompting strategy (Chain-of-Thought), model size and dataset size (Appendix A).

4 When Should We Use Human Data?

Having observed the disproportionate value added by human data, we ask what the relative cost between human and synthetic data generation must be for us to prefer one over the other. We fine-tune models on purely synthetic datasets of varying sizes, and establish the synthetic baseline by fitting a curve of the form $y = a_0 + a_1 \log(x)$ where x is the size of the synthetic dataset and y is the performance. We then take the synthetic training sets with $\{1000, 2000 \ldots\}$ points and observe the performance (y^*) when we add 200 human data points. $\exp(\frac{y^* - a_0}{a1})$ is then the size of the purely synthetic dataset that achieves equivalent performance.

Results: Across all datasets, adding **200** human data points is usually comparable to adding at least an order of magnitude (often multiple orders of magnitude) more synthetic data points. On WANLI (Figure 4), more than $17,000$ additional synthetic points are needed to achieve the performance gains of 200 human points. If the price of a synthetic point for WANLI exceeds 73 times the price of a human generated point, then an incremental amount of human annotation would be a more cost-effective way to achieve the same increase in accuracy. In the extreme case, the equation learned on FairyTaleQA suggests that it takes $2e5$ additional synthetic points to match the performance gain of 200 additional human data points. Rather than interpret these numbers literally, we take them to suggest that human data could have unique value in some settings, enabling performance levels that are impossible with purely synthetic datasets. See Appendix A for more results and details.

5 Discussion

The synthetic generations are as diverse as human data (Table 2), with comparable duplicate rates on QA datasets, and markedly fewer duplicates on most FV datasets. This is evidence that our synthetic generations are of good quality, however, even on the datasets where the synthetic data is significantly more diverse, the synthetic data does not perform as well. This suggests that diversity is an insufficient measure of quality when evaluating how good the generated data is. Our analysis shows that synthetic data generation produces claims of comparable length to the human datasets, however synthetic questions and answers tend to be longer than human-generated counterparts for all QA datasets (Figure 5). We find that synthetic generations have a higher n-gram overlap with the evidence sentences. This suggests that synthetic data generation produces data points that are more directly taken from the evidence texts, while humans are more likely to employ rephrasing or different vocabulary than the evidence texts. Surprisingly, we find that synthetic data generation chooses more varied parts of the input text as sources for the question and answer content, with human annotation overwhelmingly more likely to create questions whose answers lie at the start of the evidence texts. We include a detailed discussion in Appendix D.

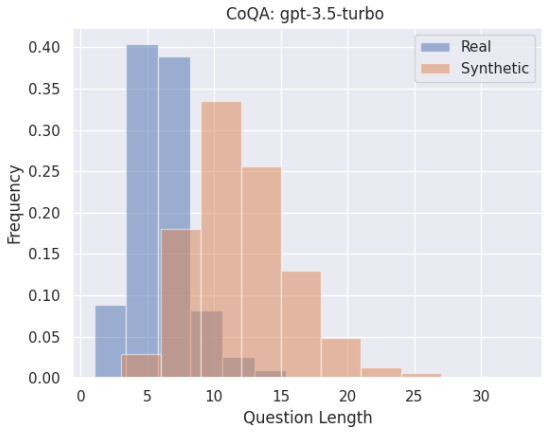


Figure 5: Synthetic questions are longer than human generated ones, a trend also seen in answers.

6 Related Work

The replacement of human annotation with synthetic data is extensively studied in the pretraining stage of LMs, where results consistently show (Shumailov et al., 2023; Seddik et al., 2024; Guo et al., 2024; Briesch et al., 2023) catastrophic forgetting, mode collapse, and performance deterioration.

In our setting, relying only on synthetic data still achieves reasonable performance across all

Dataset	Mean	Median	25th Percentile	75th Percentile
WANLI	17,671	16,905	9,711	22,931
ROPES	17,333	6,006	3,623	21,944
FairyTaleQA	281,951	36,901	15,129	813,135
FEVER	1,155	237	-1,400	7,073

Table 1: Additional synthetic data points needed to match the performance gain of 200 human data points. High values for FairyTaleQA suggest that human-generated data may unlock performance that purely synthetic data cannot achieve. Negative values for FEVER are due to a saturation of the performance gains, however, human data points reach the saturation point much faster (Appendix A)

.

Dataset	Synthetic	Human
FEVER	**5.50**	20.27
WANLI	**0.23**	1.22
SCIFACT	**0.00**	9.93
FACTIFY	**0.27**	14.93
NarrativeQA	3.85	**1.42**
CoQA	**0.54**	5.49
FairyTaleQA	2.26	**0.18**
ROPES	2.40	**1.35**

Table 2: Percentage of duplicated claims/questions for synthetic v.s. human data. Rates are comparable across datasets, but for fact verification datasets, synthetic datasets have fewer duplicates.

tasks. This suggests that the usage of exclusively synthetic data poses fewer risks when generations are grounded in diverse, natural 'evidence texts.'

Interestingly, conclusions which confirm our findings are found more in the image and multimodal domains, where recent work (Singh et al., 2024; He et al., 2023; Fan et al., 2024) finds that synthetic data holds promise, but must be used in conjunction with human data to mitigate its harms.

There is limited work on understanding whether synthetic data can replace human annotation in a task-specific setting for the language domain. Li et al. (2023) categorize text classification tasks by subjectivity, showing that synthetic data is less useful when tasks are more subjective. This draws them to focus on different tasks (sentiment classification, relation extraction and spam detection), and they do not study using a mixture of real and synthetic data. Bisbee et al. (2024) demonstrate that replacing political survey respondents with LMs produces unreliable results, while Ahmed et al. (2024) find that there are specific software engineering subtasks where synthetic data approaches human performance. Chen et al. (2024) show that instruction-following capabilities are diminished when using synthetic data and present a machine unlearning approach to mitigate this. The diversity of results when evaluating the impact of using purely synthetic data confirms that the feasibility of replacing human annotation with synthetic data is highly task dependant. This work deepens our understanding of the problem by being the first to study whether synthetic data can replace human annotation on the fundamental tasks of fact verification and evidence-based question answering.

7 Conclusion

Showing impressive performance when human data is scarce, synthetic data generation seems poised to remain a key method in FV and QA. Our work sheds light on how the best way to use this method is in conjunction with human data. We show that a little human data goes a long way, with just **125** points being enough to see reliable gains on all datasets studied. With practical considerations in mind, we show that the alternative to small amounts of additional human data can be an order of magnitude more of synthetic data, suggesting that at times human annotation can be cost-effective relative to synthetic generation. We hope these results better inform design decisions on datasets and methods for fact verification and question answering.

8 Limitations

While we include results on multilingual Fact Verification datasets, the primary focus of our work is limited to the English language. Additionally, our results on multilingual datasets suggest that while similar claims can be made regarding the impact of replacing human annotation with synthetic data across different languages, the amount of human data needed to observe a meaningful performance increase may vary across languages. We also have a limited ability to control for dataset leakage, with

only one dataset from each of the tasks that is surely not leaked to GPT-3.5 (and, even these two datasets may have been seen by GPT-4). This can potentially bias the results in favor of synthetic data. Due to the scarcity of suitable available datasets (i.e., ones that have not been exposed to the prompt models) we are prevented from studying the problem more rigorously. Another limitation is that while we are able to identify clear differences between synthetic vs. real data distributions, our analysis of the errors made by models trained on 0% vs. 100% synthetic data failed to yield any generalizable insights that could inform modelling approaches. A more fine-grained study of the effect of using synthetic data on the behaviour of the downstream model is hence left as a subject of future research.

9 Ethical Considerations

The usage of synthetic data has several important ethical considerations. In the era of LMs trained on internet-wide corpora having poor documentation as to their exact data sources, it becomes challenging to ensure the privacy of individuals whose data may be obtainable via a public crawl (Yao et al., 2024). Additionally, models trained on massive internet-based data sources may contain implicit biases, as well as illegal and/or highly offensive material that is hard to audit and clean (Bender et al., 2021). This data affects the synthetic data obtained from prompt models, and could unknowingly impose cultural or ethical viewpoints that are unintended or not well aligned with the use case in mind. Specifically, prior work has shown that one of the prompt models studied in this work, GPT-3.5, often disagrees with humans on key ethical questions (Felkner et al., 2024). The endeavour to completely replace human annotation with synthetic data generation also has key implications on the extent to which the field of NLP employs human annotators. It is possible that an increasing reliance on purely synthetic data reduces the demand for human annotation, which would place a downward pressure on the working standards and compensation awarded to the remaining human annotators (Weidinger et al., 2022). We argue in this work that we should not try to eliminate human annotation from our dataset and method design, showing that their work contributes uniquely helpful data points.

10 Acknowledgements

This work was funded by the Defense Advanced Research Projects Agency with award HR00112220046. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the view of our sponsors.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Toufique Ahmed, Prem Devanbu, Christoph Treude, and Michael Pradel. 2024. Can llms replace manual annotation of software engineering artifacts? *ArXiv*, abs/2408.05534.

Dhananjay Ashok, Atharva Kulkarni, Hai Pham, and Barnabas Poczos. 2023. The student becomes the master: Outperforming GPT3 on scientific factual error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6762–6778, Singapore. Association for Computational Linguistics.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

James Bisbee, Joshua D. Clinton, Cassy Dorff, Brenton Kenkel, and Jennifer M. Larson. 2024. Synthetic replacements for human survey data? the perils of large language models. *Political Analysis*.

Martin Briesch, Dominik Sobania, and Franz Rothlauf. 2023. Large language models suffer from their own output: An analysis of the self-consuming training loop. *ArXiv*, abs/2311.16822.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jie Chen, Yupeng Zhang, Bingning Wang, Xin Zhao, Ji-Rong Wen, and Weipeng Chen. 2024. Unveiling the flaws: Exploring imperfections in synthetic data and mitigation strategies for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14855–14865, Miami, Florida, USA. Association for Computational Linguistics.

Mei-Hua Chen, Shih-Ting Huang, Chung-Chi Huang, Hsien-Chin Liou, and Jason S Chang. 2012. Prefer: using a graph-based approach to generate paraphrases for language learning. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 80–85.

Adria De Gispert, José B Mariño, and Josep Maria Crego. 2005. Improving statistical machine translation by classifying and generalizing inflected verb forms. In *INTERSPEECH*, pages 3193–3196.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.

Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. 2024. Scaling laws of synthetic images for model training... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7382–7392.

Virginia Felkner, Jennifer Thompson, and Jonathan May. 2024. GPT is not an annotator: The necessity of human annotation in fairness benchmark construction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14104–14115, Bangkok, Thailand. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *ArXiv*, abs/2312.10997.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. 2024. The curious decline of linguistic diversity: Training language models on synthetic text. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3589–3604, Mexico City, Mexico. Association for Computational Linguistics.

Hasan Hammoud, Hani Itani, Fabio Pizzati, Philip H. S. Torr, Adel Bibi, and Bernard Ghanem. 2024. Synthclip: Are we ready for a fully synthetic clip training? *ArXiv*, abs/2402.01832.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. 2023. Is synthetic data from generative models ready for image recognition? In *The Eleventh International Conference on Learning Representations*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Mohammad Javad Hosseini, Andrey Petrov, Alex Fabrikant, and Annie Louis. 2024. A synthetic data approach for domain generalization of NLI models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2212–2226, Bangkok, Thailand. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason E Weston, and Mike Lewis. 2024. Self-alignment with instruction backtranslation. In *The Twelfth International Conference on Learning Representations*.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language

models for text classification: Potential and limitations. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 58–62, Hong Kong, China. Association for Computational Linguistics.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. Wanli: Worker and ai collaboration for natural language inference dataset creation. In *Conference on Empirical Methods in Natural Language Processing*.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477.

Shreyash Mishra, S Suryavardan, Amrit Bhaskar, Parul Chopra, Aishwarya N. Reganti, Parth Patwa, Amitava Das, Tanmoy Chakraborty, A. Sheth, and Asif Ekbal. 2022. Factify: A multi-modal fact verification dataset. In *DE-FACTIFY@AAAI*.

Jingwei Ni, Minjing Shi, Dominik Stammbach, Mrinmaya Sachan, Elliott Ash, and Markus Leippold. 2024. Afacta: Assisting the annotation of factual claim detection with reliable llm annotators. In *Annual Meeting of the Association for Computational Linguistics*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Tobias Schimanski, Jingwei Ni, Mathias Kraus, Elliott Ash, and Markus Leippold. 2024. Towards faithful and robust LLM specialists for evidence-based question-answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1913–1931, Bangkok, Thailand. Association for Computational Linguistics.

Mohamed El Amine Seddik, Suei-Wen Chen, Soufiane Hayou, Pierre Youssef, and Mérouane Debbah. 2024. How bad is training on synthetic data? a statistical analysis of language model collapse. *ArXiv*, abs/2404.05090.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Umang Sharma, Sidarth Saran, and Dr Shankar M. Patil. 2023. Fake news detection using machine learning algorithms. *2023 International Conference on New Frontiers in Communication, Automation, Management and Security (ICCAMS)*, 1:1–7.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *ArXiv*, abs/2305.17493.

Patrice Y Simard, Yann A LeCun, John S Denker, and Bernard Victorri. 2002. Transformation invariance in pattern recognition—tangent distance and tangent propagation. In *Neural networks: tricks of the trade*, pages 239–274. Springer.

Krishnakant Singh, Thanush Navaratnam, Jannik Holmer, Simone Schaub-Meyer, and Stefan Roth. 2024. Is synthetic data all we need? benchmarking the robustness of models trained with synthetic images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2505–2515.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957, Miami, Florida, USA. Association for Computational Linguistics.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. Minicheck: Efficient fact-checking of llms on grounding documents. *ArXiv*, abs/2404.10774.

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining? *ArXiv*, abs/2303.04360.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 1 (Long Papers), pages 809–819.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Position: Will we run out of data? limits of LLM scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Yuwei Wan, Yixuan Liu, Aswathy Ajith, Clara Grazian, Bram Hoex, Wenjie Zhang, Chunyu Kit, Tong Xie, and Ian Foster. 2024. Sciqag: A framework for auto-generated science question answering dataset with fine-grained evaluation.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? GPT-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205, Punta Cana, Dominican Republic. Association for Computational Linguistics.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563, Lisbon, Portugal. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, et al. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 214–229.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv: Artificial Intelligence*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Wenhao Liu, Pascale Fung, and Caiming Xiong. 2022. QAConv: Question answering on informative conversations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5389–5411, Dublin, Ireland. Association for Computational Linguistics.

Ziang Xie, Sida I. Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y. Ng. 2017. Data noising as smoothing in neural network language models. In *International Conference on Learning Representations*.

Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Jia-Jun Li, Nora Bradford, Branda Sun, Tran Bao Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022. Fantastic questions and where to find them: FairytaleQA – an authentic dataset for narrative comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 447–460, Dublin, Ireland. Association for Computational Linguistics.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, page 100211.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Supplemental Figures

We present a detailed set of figures and tables to supplement the results presented in the main text.

Main Experiments: For figures in the main text where only one task is shown (Figure 1 and Figure 2), we provide the complete figures with both tasks (Figure 6 and Figure 7). We also provide the individual performance curves for these experiments (Figure 8 and Figure 10).

Robustness to choice of QA metric: To verify the robustness of the results, we show that the QA results are not an artifact of the choice of metric (Figure 3 and Figure 4) by using Exact Match, String Inclusion, ROUGE-1 (Lin, 2004) and BERTScore (Zhang et al., 2020). There is overwhelming agreement between all metrics on the rankings of models.

Addressing spurious correlations: We show that the performance gains afforded by human generated data cannot be explained by a spurious correlation between the human generated train and test splits. This would occur when there are significant annotation artifacts that are not relevant to the task, but are correlated with the correct output. We conduct an out-of-domain experiment (Table 5), using different datasets to source the training data and testing on a single hold out dataset. Using more synthetic data leads to performance declines even in the OOD setting, showing that human data is of higher quality and the results from the main text cannot be explained by a spurious correlation between the human test and human training samples. Interestingly, in the OOD setting the decline is steady, and we do not observe the phenomenon of a small amount of human data having a disproportionate impact on performance. This suggests that the disproportionate impact of human data occurs when the human data is in-domain. We leave a further exploration of the OOD generalization abilities of synthetic vs. human data to future work.

Multilingual Experiments: We replicate our experiment using the Arabic, Georgian, and Indonesian splits of the XFact dataset. We observe (Figure 9) the same trend as those from earlier experiments, confirming that our results are not limited to the English language. While the phenomenon is reproduced, the threshold of replacement at which we observe a precipitous decline is not the same across languages. We hypothesize that the language-specific threshold at which a little human data leads to significant performance increases is dependent on how low resource the language is. The study of synthetic data in the multilingual setting has unique considerations that we have not addressed in this work; we leave a focus on these problems to future work.

Ablations: We show that the same trends can be seen (Figures 11 to 15) when using a different fine-tuning model (Mistral-7B), models of varying scales (from 1B parameter models to 30B parameter models. different prompting models (GPT-4 and Claude-3.5) and a more sophisticated prompting strategy (Chain-of-Thought Prompting). Across all configurations, we see a consistent decrease in performance when moving from 95% to 100% synthetic data, confirming that models trained on purely synthetic data can be improved by including just **125** real data points. For Chain-of-Thought Prompting, the authors manually annotated 3 examples with rationales per dataset to serve as the prompts. The complete examples and pipeline are provided with the code: `github.com/dhananjayashok/littlehumandata`

We additionally show that these trends hold across data scales (Figures 16 and 17), replicating the experiment with n=3000 and n=1000. While the trend is clearly visible in both cases, the results for n=1000 have more variance and hence have a minority of cases where the relationship does not hold.

Tradeoff Experiment: The main text shows results for the experiment detailed in Section 4 on the WANLI dataset (Figure 4), here we show results on the remaining three datasets (Figure 18) and provide (Figure 1) the number of additional synthetic points needed to match the performance gains of 200 additional real points (average, median and standard deviation for each dataset). ROPES shows similar results to WANLI, however FairyTaleQA and FEVER present different trends. On FEVER, we are able to reach the saturation point, after which additional data (whether synthetic or real) does not increase performance. Even in this case, we are able to reach this point of diminishing marginal return more rapidly when using a small amount of synthetic data. On a base synthetic training set of size 3000, adding 200 real data points drives the test accuracy to 89.25%, a score that is only matched once we add at least 2000 synthetic data points (an order of magnitude larger). On FairyTaleQA, we get enormous estimates for the number of additional synthetic points needed (a mean of 2.8e5). We do not interpret these numbers

literally, rather seeing this as a sign that human generated data may occasionally boost performance to an extent that could be fundamentally unachievable by purely synthetic data.

B Synthetic Data Generation

In our implementation (Figure 19), we use few-shot learning with $k = 3$, i.e., three examples per query, with each example drawn randomly (with replacement) from the training set of the specific dataset.

We generate one synthetic point for every real point in the dataset, using the evidence text it is associated with. This gives us a total of n synthetic data points for every n real data points in a dataset.

We observed that if we did not correct for label shift, the prompt model would be heavily biased towards True claims, i.e., it would generate a dataset containing 90% True claims, while original datasets have proportions between 33%–60% True.

For the synthetic datasets used in our experiments, we correct for this label shift by specifying the label of the claim we wish to generate and providing only examples of claims with that specific label in the prompt.

For all datasets, we verify that the diversity of the generated claims/questions/answers are comparable to that of the human generated texts (see Appendix D)

This setting is generous towards synthetic data generation. In practice, we might only have three fixed examples to use in the prompt, potentially reducing the diversity of synthetic data generated. We verify that this does not affect the results of our experiment in the Chain-of-Thought ablation (Figure 15), where we use a fixed set of examples to generate all synthetic data points. We may also not know the correct label proportion to ask for and suffer a significant label shift when using synthetic data generation.

C Datasets Used

All datasets used below are released under open use licenses, authorizing their use in this research. For each dataset, we discuss the potential of dataset leakage (i.e., whether the data has been exposed to GPT-3.5-Turbo during its training) as well as the extent of automation involved in the generation of each dataset. However, across all experiments and ablations, these factors do not seem to have any discernable effect on the trends discovered in this work.

C.1 Fact Verification Datasets

FEVER (Thorne et al., 2018) is a dataset of claims about specific entities, generated by altering sentences extracted from Wikipedia. The evidence passages are sentences from Wikipedia articles relevant to the entity in question. This dataset has been well established for a long time before the release of the prompting models used in this work, increasing the chance that it has been exposed to the prompt model ahead of time.

SciFact (Wadden et al., 2020) is a fact verification dataset for the scientific domain, which uses the abstracts of scientific articles as evidence texts. The corpus is collected from S2ORC (Lo et al., 2020), a publicly-available corpus of millions of scientific articles. Annotators are shown a source citation in the context of an article, and are asked to write up to three claims based on the content of the citation.

The above datasets are popular NLP challenge sets that were well known even before the release of GPT-3.5-Turbo (Brown et al., 2020), the prompting model used in this work. The following two datasets were released after the official training date cut-off, guaranteeing that the data has not been seen ahead of time.

WANLI (Liu et al., 2022) is an NLI dataset of 108K examples created through a hybrid worker and AI collaboration approach. The creators first study MultiNLI (Williams et al., 2018) and use dataset cartography to automatically identify examples that demonstrate challenging reasoning patterns. and then instruct GPT-3 to compose new examples with similar patterns. Machine generated examples are then automatically filtered, and finally revised and labeled by human crowd workers. While GPT-3.5-Turbo has not been trained on this data, it is worth noting that the data is partially synthetically generated.

FACTIFY (Mishra et al., 2022) is a dataset on multi-modal fact verification. It contains images, textual claims, reference textual documents and reference images. The dataset marks some examples that can be verified using text only; we use this sample in our experiments. This dataset was released after the training cut-off date for GPT-3.5 and takes its evidence texts/claims from human-written news or editorial articles. This ensures that the prompt models studied have not seen the data before training.

Label mapping for NLI and FV: While all of the above datasets contain labels for Supports, Refutes, and Not Enough Information (or Entails, Contradicts, Neutral), we consider the stricter formulation of Fact Verification used by Honovich et al. (2022) and Zha et al. (2023), considering a claim to be factual if the label is Supports (Entails), and non-factual otherwise.

C.2 Question Answering Datasets

ROPES (Lin et al., 2019) is a QA dataset which tests a system's ability to apply knowledge from a passage of text to a new situation. The evidence context contains causal or qualitative relation(s) (e.g., "animal pollinators increase efficiency of fertilization in flowers"), and a novel situation that uses this background. The question requires reasoning about effects of the relationships in the background passage in the context of the situation.

CoQA (Reddy et al., 2019) is a dataset for building Conversational Question Answering systems. CoQA measures the ability of machines to understand a text passage and answer a series of interconnected questions that appear in a conversation. In our experiments, we extract only the first question in the series and use this to obtain our (context, question, answer) data points.

QAConv (Wu et al., 2022) focuses on informative conversations, including business emails, panel discussions, and work channels. The creators collect QA pairs with both human-written and machine-generated questions. They use a question generator and a dialogue summarizer as auxiliary tools to collect and recommend questions. While the arXiv version of the paper appears before the GPT-3 cut-off data (April 2021 to the cut-off date of Sept 2021), the paper itself appeared only at ACL 2022. It is still possible that the training data was compromised, and owing to the lack of clarity on the training data used for GPT-3 we have no way to confirm or deny this speculation.

FairyTaleQA (Xu et al., 2022) is a dataset focusing on narrative comprehension of kindergarten to eighth-grade students. The evidence texts are derived from children-friendly stories which serve as evidence texts. The questions are both explicit and implicit, covering seven types of narrative elements or relations. This dataset was released after the GPT-3 training cut-off date, ensuring that it has not been seen by our prompt model before.

D Detailed Discussion on Differences Between Synthetic and Human Data

To compute the extent to which the evidence sentences 'contain' the questions, answers, and claims, we measure the BLEU of the generation with **each individual sentence** of the evidence texts, plotting the maximum of these BLEU scores in Figure 21. We find that synthetic generations have a far higher n-gram overlap with the evidence sentences than human generations. This suggests that synthetic data generation produces data points that are more extractive, while humans are more likely to abstract from the evidence. We also use the position of the evidence sentence that achieves the highest BLEU score as a proxy for the source location of the synthetic generation, and find that synthetic data generation chooses more diverse sources for the question and answer content, with human annotation overwhelmingly more likely to create questions whose answers lie in the start of the evidence texts (Figure 22). Finally, the main text shows the size length comparison for a single dataset. Here we provide a larger sample (Figure 20). We explore the errors created by the models trained on 0% and 100% data, searching for trends or divergences between the input instances that achieve a low prediction accuracy or score. Our investigation finds no major distinguishing factors between them, leaving a more fine-grained study of the effect of purely synthetic data on model decision-making to future work.

E Implementation Details

While our full code implementation can be seen in the GitHub repository (to be released after review), we list the key implementation details below.

Hardware and Systems Used: The experiments were run on a cluster that included nodes with: five A40 GPUs (48GB), three RTX 2080Tis, and a separate machine using a single A100 GPU.

Prompt Models used: We used GPT-3.5-Turbo and GPT-4-Turbo Batch APIs from OpenAI. Generations were obtained at various points from August 2024 to September 2024.

Fine-Tuning Models Used: We used two fine-tuning models in our experiments. Llama3 used the Llama3.1-8B HuggingFace Checkpoint, and Mistral used the Mistral7B-Instruct-v0.2 HuggingFace Checkpoint. We did not conduct an extensive hyperparameter search, however we tried various epochs on smaller samples of the FEVER and

ROPES datasets, selecting that number for every dataset on all experiments. Fact verification models used Adam Optimization with a learning rate of 1e-5 for two epochs, while QA datasets used a learning rate of 1e-2 for five epochs.

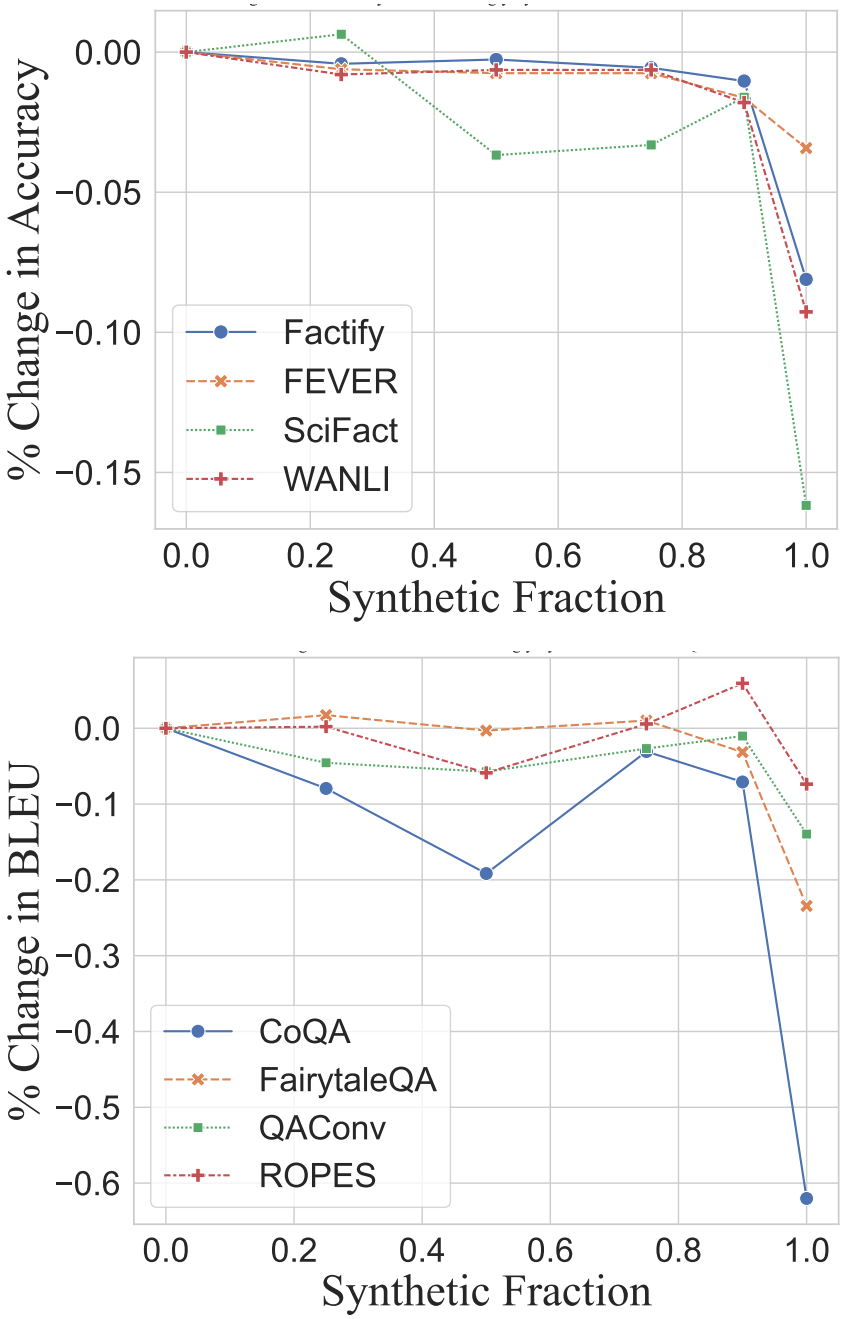


Figure 6: Change in model performance as the proportion of synthetic points in the training data is increased. Across datasets, the performance decrease when moving from 0% to 90% synthetic data is often less than that of moving from 90% to purely synthetic data.

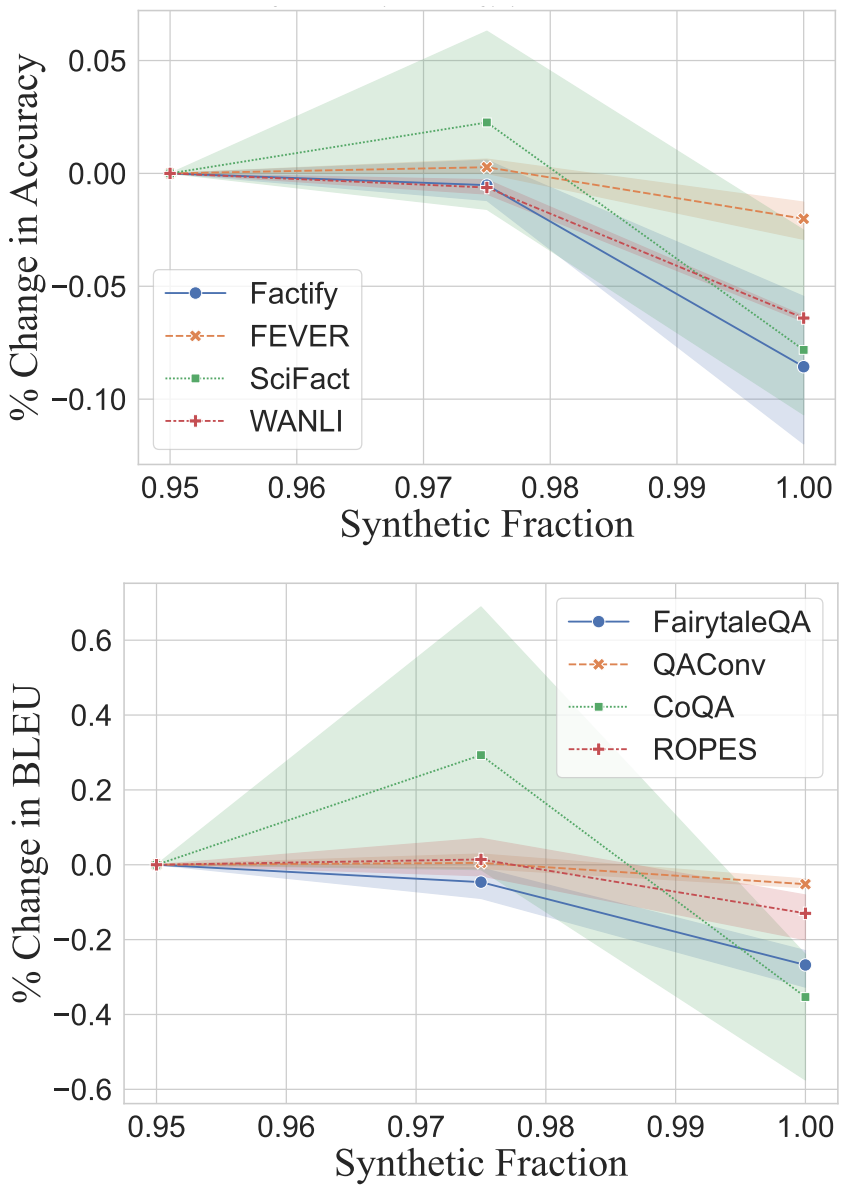


Figure 7: Model performance as the synthetic proportion of the training data varies from 95% to 100%. Across all datasets and random seeds, having **just 2.5%** of the training dataset being human generated boosts performance.

Dataset	Synthetic %	EM	Inc	R Inc	BLEU	ROUGE	BERTScore
CoQA	0	40.6	52.2	60.8	47.9	64.0	87.8
	25	35.0	51.8	54.8	44.1	62.0	85.9
	50	31.6	42.8	60.6	38.7	54.7	79.9
	75	39.2	50.4	69.0	46.4	62.1	79.9
	90	36.2	50.2	58.2	44.5	60.8	85.4
	100	13.6	26.0	58.2	18.2	26.6	52.8
FairytaleQA	0	0.0	0.0	0.0	39.3	55.3	90.8
	25	0.0	0.0	0.0	40.0	56.1	90.5
	50	0.0	0.0	0.0	39.2	55.3	88.9
	75	0.0	0.0	0.0	39.7	55.4	90.6
	90	0.0	0.0	0.0	38.1	54.2	89.9
	100	0.0	0.0	0.0	30.1	49.5	88.4
QAConv	0	29.4	36.3	49.4	35.1	51.6	89.9
	25	28.6	34.5	48.0	33.5	48.7	89.6
	50	28.0	34.0	47.1	33.1	48.7	89.5
	75	28.6	35.9	48.5	34.2	50.5	90.0
	90	29.0	36.0	49.9	34.8	50.6	89.0
	100	23.5	34.3	41.0	30.2	45.2	87.2
ROPES	0	66.8	67.4	72.2	67.0	72.7	96.0
	25	66.8	67.5	69.8	67.2	71.1	96.2
	50	62.8	63.4	65.5	63.1	66.6	95.2
	75	66.8	68.0	68.8	67.4	70.8	96.2
	90	70.6	71.5	71.8	71.0	73.0	96.8
	100	60.8	63.9	61.2	62.1	64.9	95.3

Table 3: Full Results for the QA datasets. There is overwhelming agreement between all metrics on the ranking between models trained on different synthetic fractions. EM: Exact Match, Inc: String Inclusion, R Inc: Reverse String Inclusion

Run	Dataset	Synthetic %	BLEU	ROUGE	BERTScore
0		95	38.78	54.80	90.34
	FairytaleQA	97.5	37.19	52.09	86.17
		100	26.10	43.82	77.03
		95	34.45	51.23	90.35
	QAConv	97.5	34.31	51.89	89.80
		100	32.33	49.39	89.77
		95	25.33	35.64	59.39
	CoQA	97.5	42.78	57.88	78.84
		100	19.11	27.57	51.44
		95	72.89	77.26	97.22
	ROPES	97.5	70.74	73.83	96.58
		100	58.28	60.82	94.37
1		95	39.50	54.67	90.57
	FairytaleQA	97.5	35.95	53.10	89.73
		100	29.75	47.05	81.86
		95	38.94	56.41	90.79
	QAConv	97.5	40.06	57.64	90.45
		100	37.46	54.11	88.57
		95	34.84	47.16	63.35
	CoQA	97.5	41.91	56.53	85.30
		100	14.81	24.58	50.02
		95	69.61	71.97	96.23
	ROPES	97.5	69.72	72.80	96.58
		100	62.08	65.44	95.03
2		95	37.97	53.98	90.34
	FairytaleQA	97.5	37.65	52.44	87.83
		100	29.26	49.70	88.56
		95	38.07	54.49	90.13
	QAConv	97.5	37.70	54.64	88.61
		100	35.94	51.84	89.49
		95	30.80	42.03	62.72
	CoQA	97.5	30.40	41.02	56.83
		100	23.42	37.20	63.31
		95	63.46	65.80	95.28
	ROPES	97.5	67.94	71.47	96.16
		100	58.34	61.62	94.16

Table 4: Results on $n = 5000$ from 95% to 100% for the QA datasets. There is overwhelming agreement between all metrics on the ranking between models trained on different synthetic fractions.

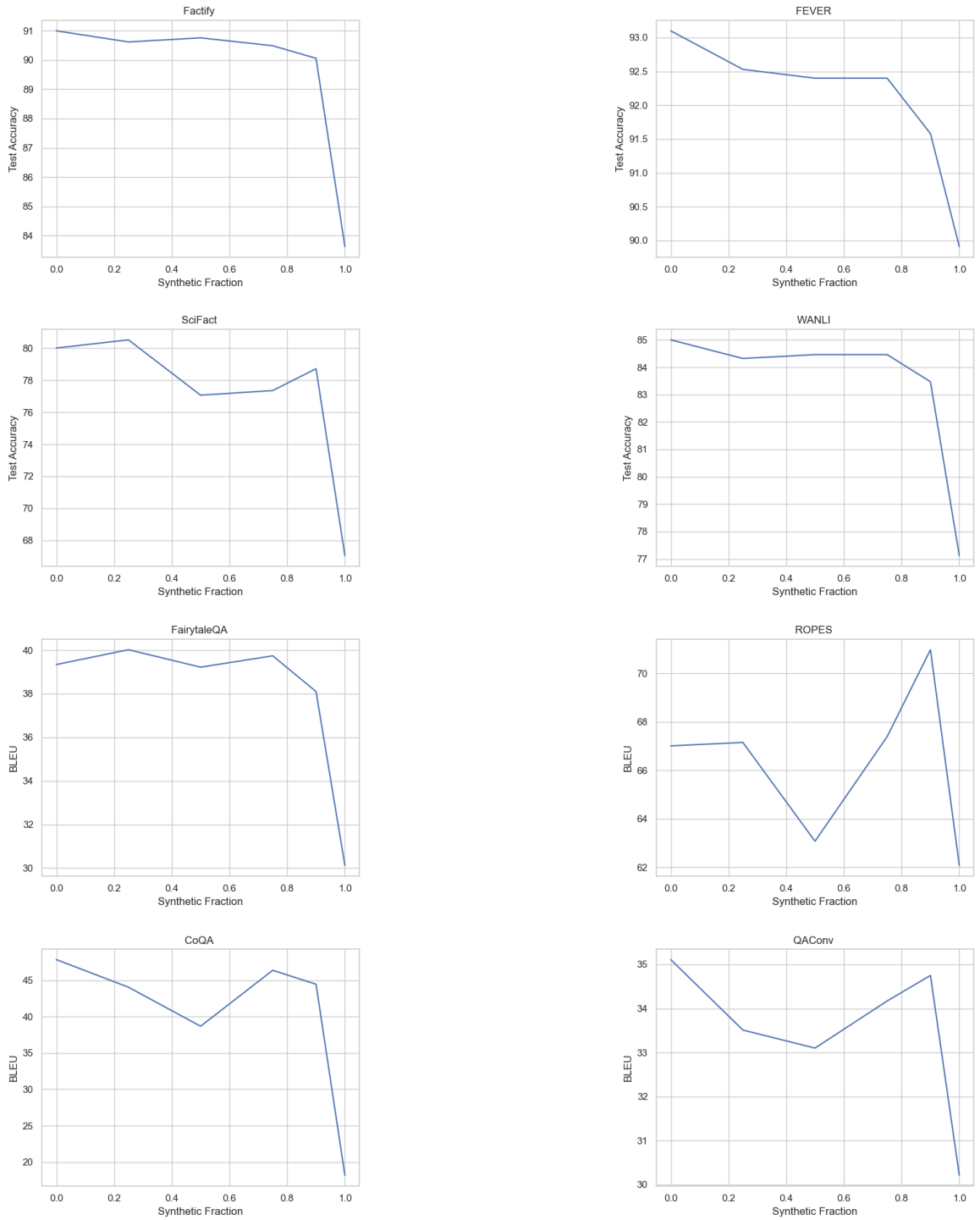


Figure 8: Change in model performance as the proportion of synthetic points in the training data is varied. Across datasets, the performance decrease when moving from 0% to 90% synthetic data is often less than that of moving from 90% to purely synthetic data.

Train Sets	Test Set	Synthetic %	Test Accuracy
		0	69.98
		25	67.56
FEVER, SciFact	WANLI	50	65.86
		75	64.82
		100	64.36
		0	83.01
		25	80.64
WANLI, SciFact	FEVER	50	79.22
		75	78.94
		100	76.01
		0	71.76
		25	69.75
FEVER, WANLI	SciFact	50	69.82
		75	66.42
		100	64.41

Table 5: Test accuracy when replacing human data with synthetic data in the out-of-distribution setting. Using more synthetic data leads to performance declines even in the OOD setting, showing that human data is of higher quality and the results from the main text cannot be explained by a spurious correlation between the human test and human training samples.

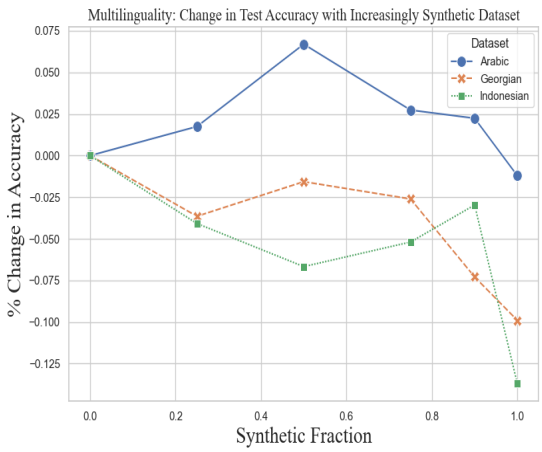


Figure 9: Change in model performance as the proportion of synthetic points in the training data is increased on multilingual fact verification datasets (splits of X-Fact). We observe the same trend as those from earlier experiments, confirming that our results are not limited to the English language. While the phenomenon is reproduced, the threshold of replacement at which we observe a precipitous decline is not the same across languages.

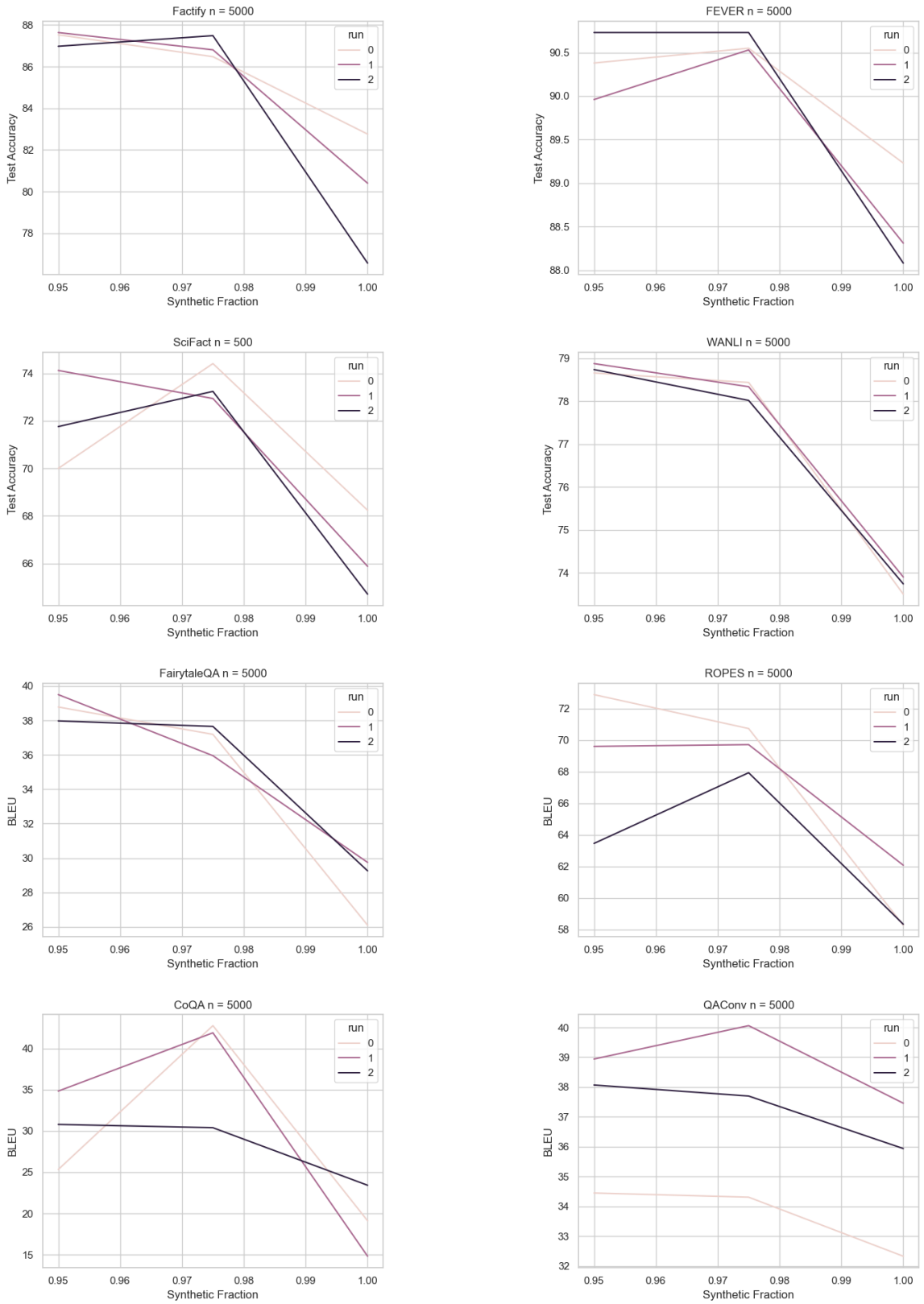


Figure 10: Model performance as the synthetic proportion of the training data varies from 95% to 100%. Across all datasets and random seeds, having **just 2.5%** of the training dataset being human generated boosts performance.

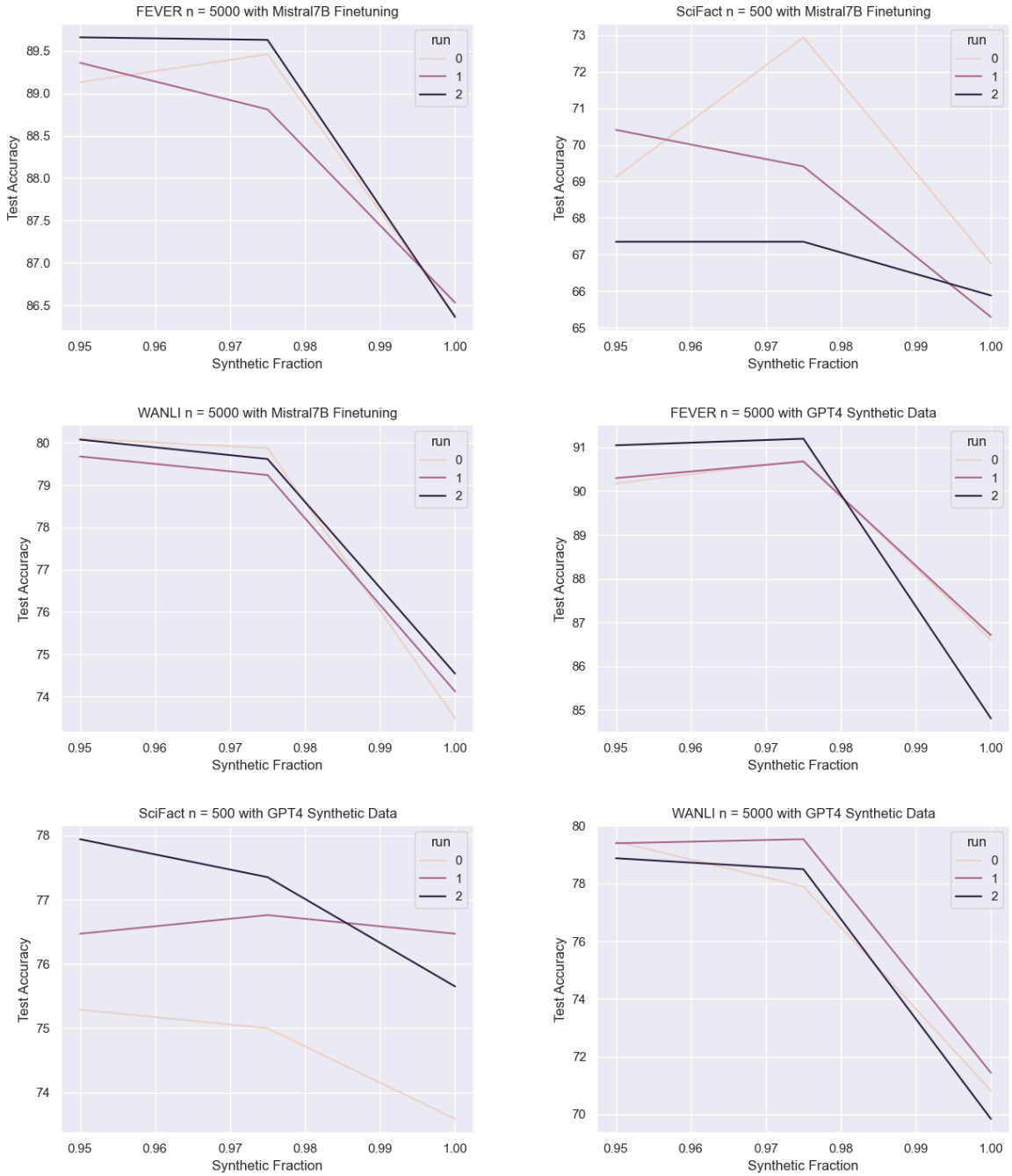


Figure 11: Results hold consistently on Fact Verification datasets when using Mistral7B as the fine-tuning model and GPT-4 as the prompting model.

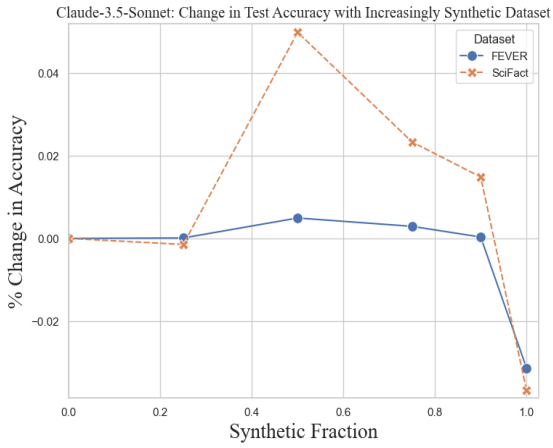


Figure 12: Results hold when using Claude-3.5-Sonnet as the prompting model, showing that the phenomenon is not particular to Synthetic Data from GPT based models.

Dataset	Claim / Question	
	Synthetic	Human
FEVER	35.78	42.76
WANLI	15.15	20.10
SCIFACT	7.12	20.92
FACTIFY	14.50	23.93
NarrativeQA	30.92	8.25
CoQA	7.08	8.39
FairyTaleQA	22.59	16.85
ROPES	28.73	41.42

Table 6: 4-Gram overlap % between all synthetic and human generated claims / questions for each dataset. On several datasets, synthetic claims have a lower overlap

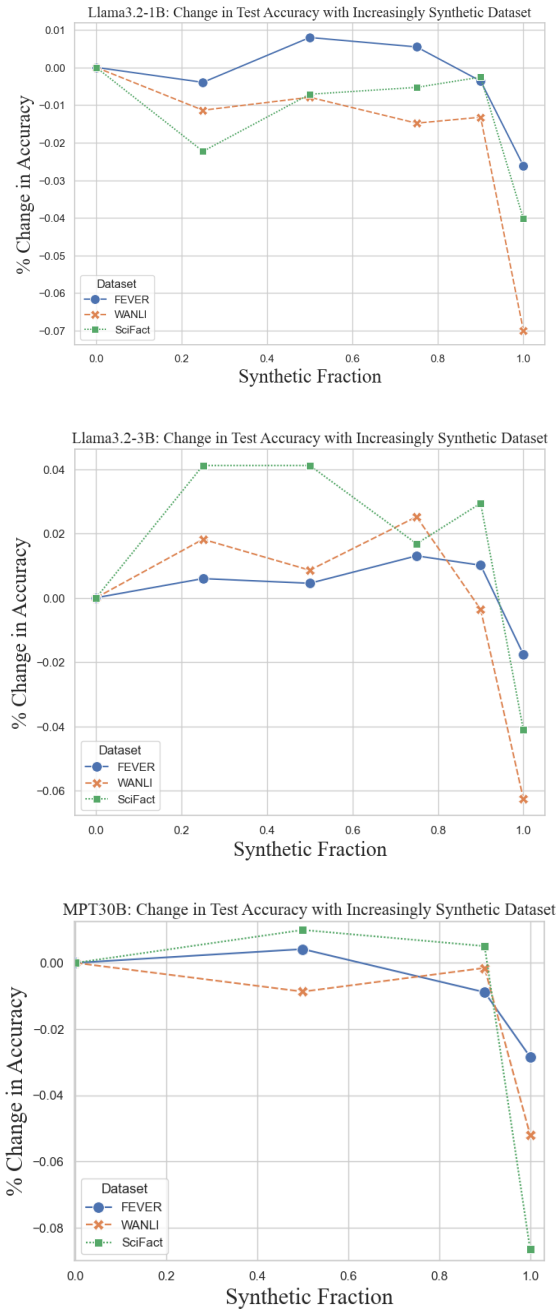


Figure 13: Results hold consistently on Fact Verification datasets when using models of different scales.

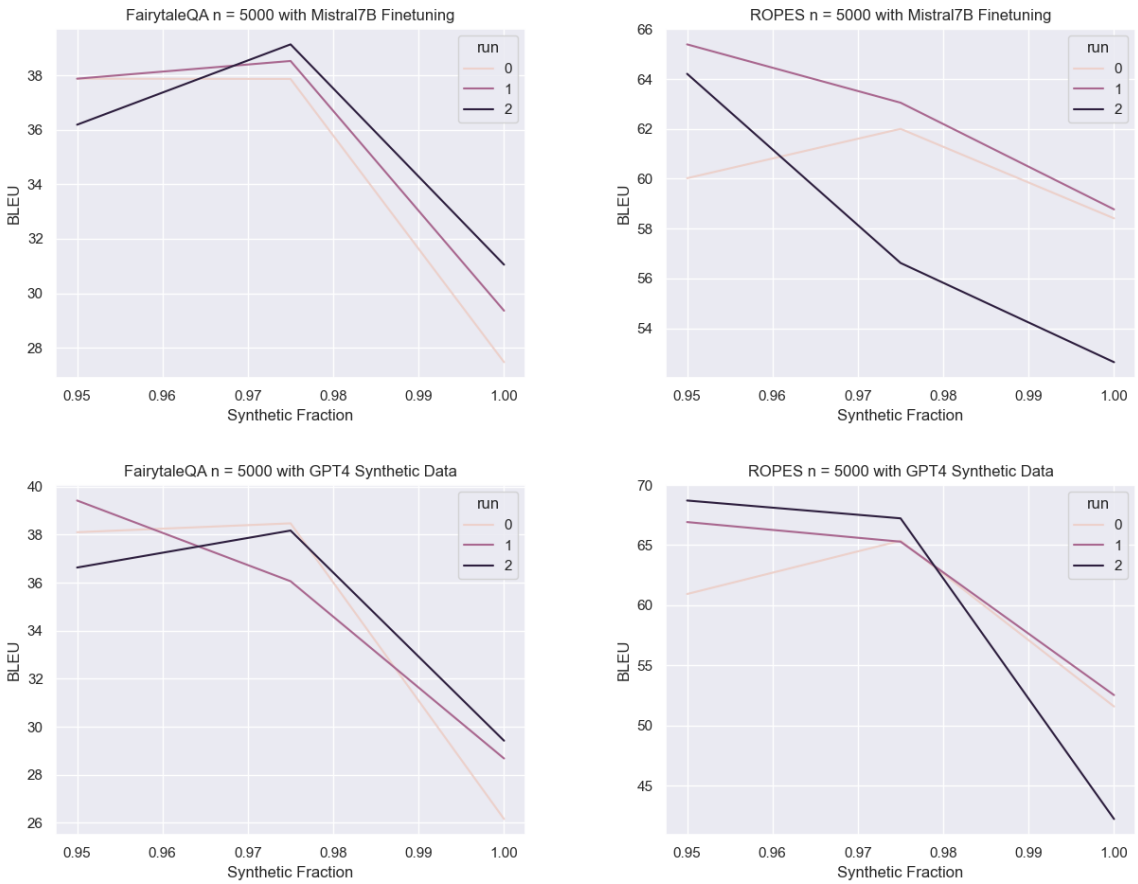


Figure 14: Results hold consistently on Question Answering datasets when using Mistral7B as the fine-tuning model and GPT-4 as the prompting model

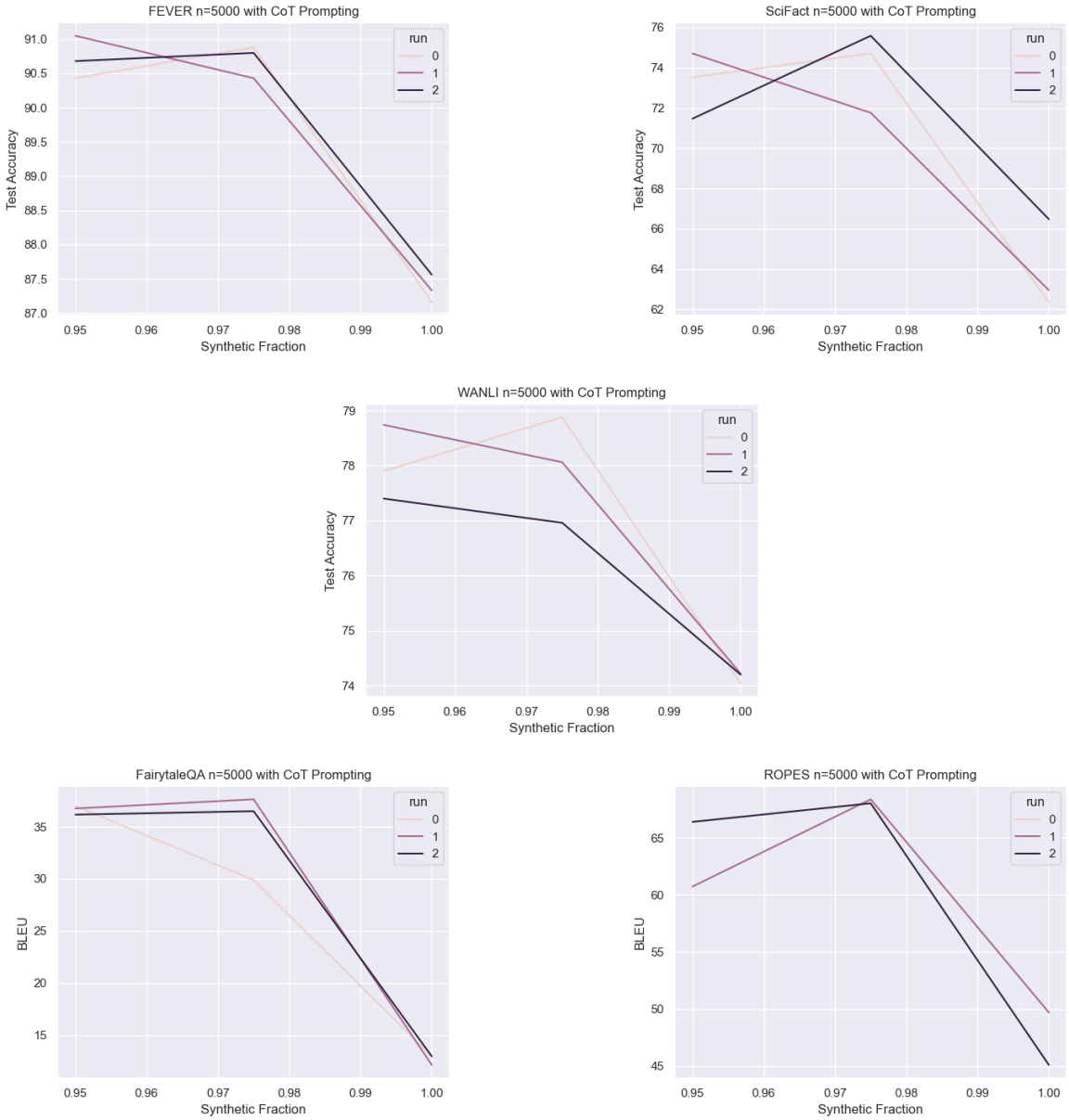


Figure 15: Results hold when using Chain-Of-Thought Prompting on GPT-3.5

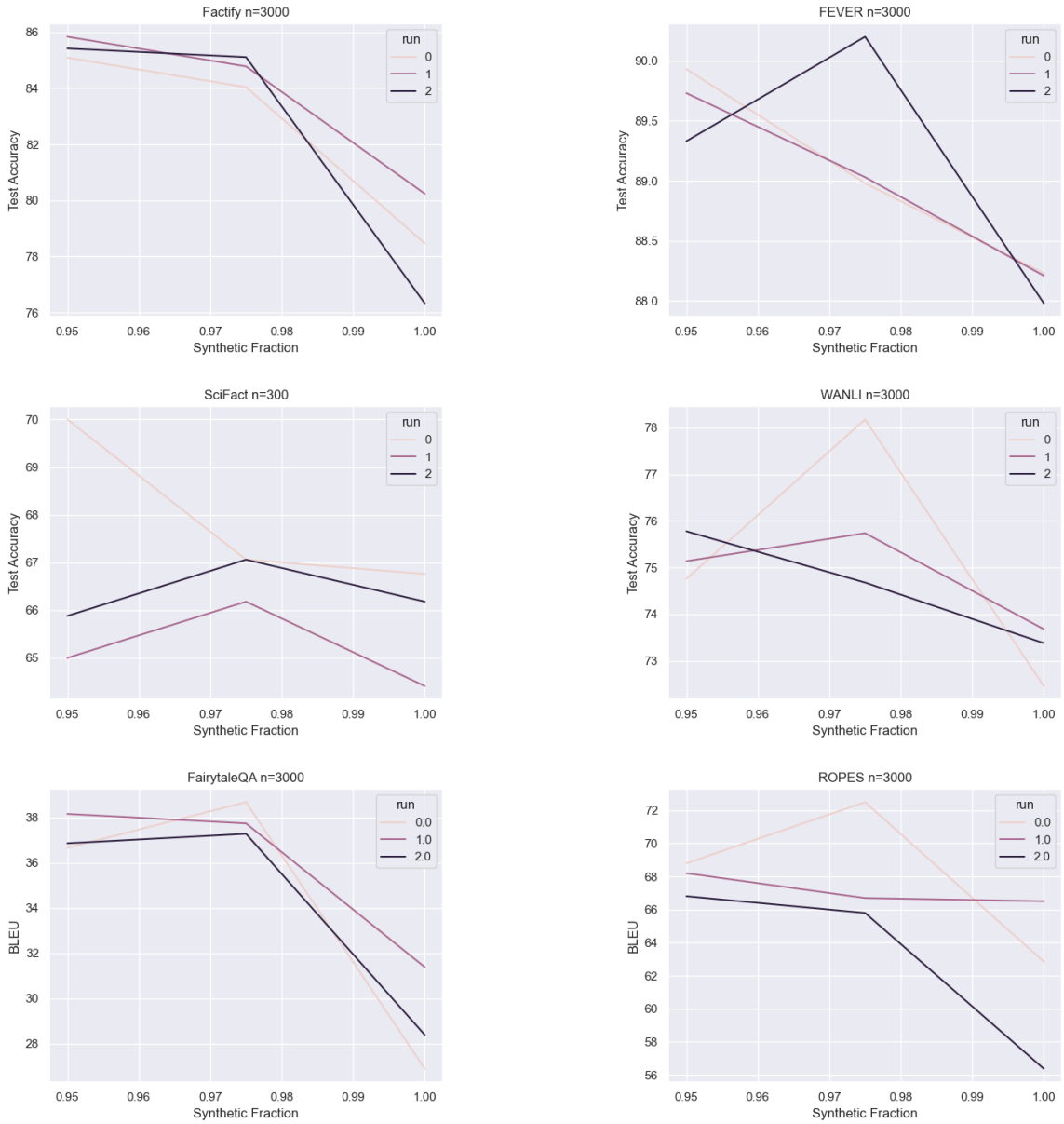


Figure 16: Model performance as the synthetic proportion of the training data varies from 95% to 100% with total number of points $n = 3000$. Across all runs on all datasets including **just 75 real datapoints** can boost performance.

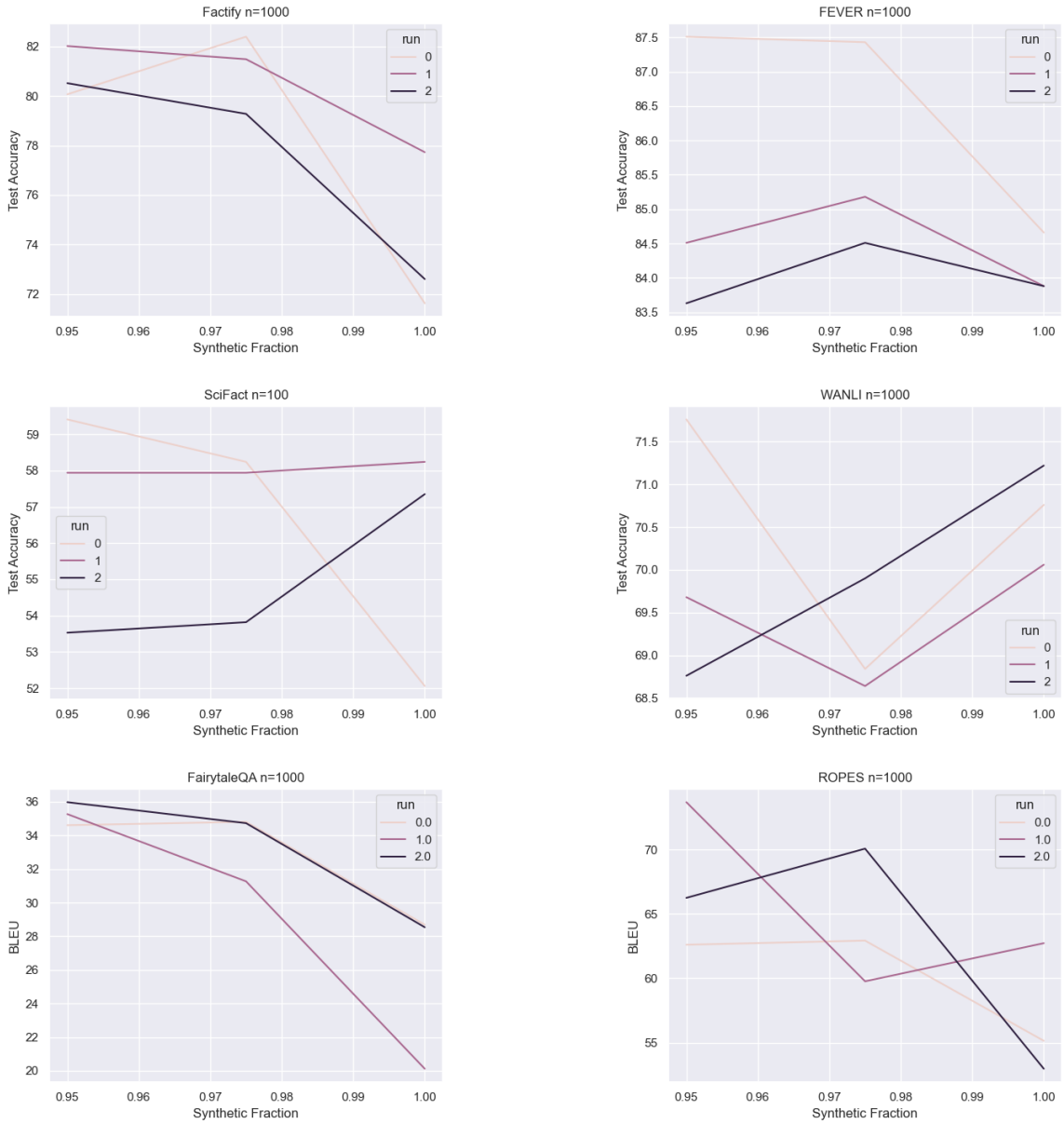


Figure 17: Model performance as the synthetic proportion of the training data varies from 95% to 100% with total number of points $n = 1000$. While the most common trend is that including real data improves performance, the results are much more unstable.

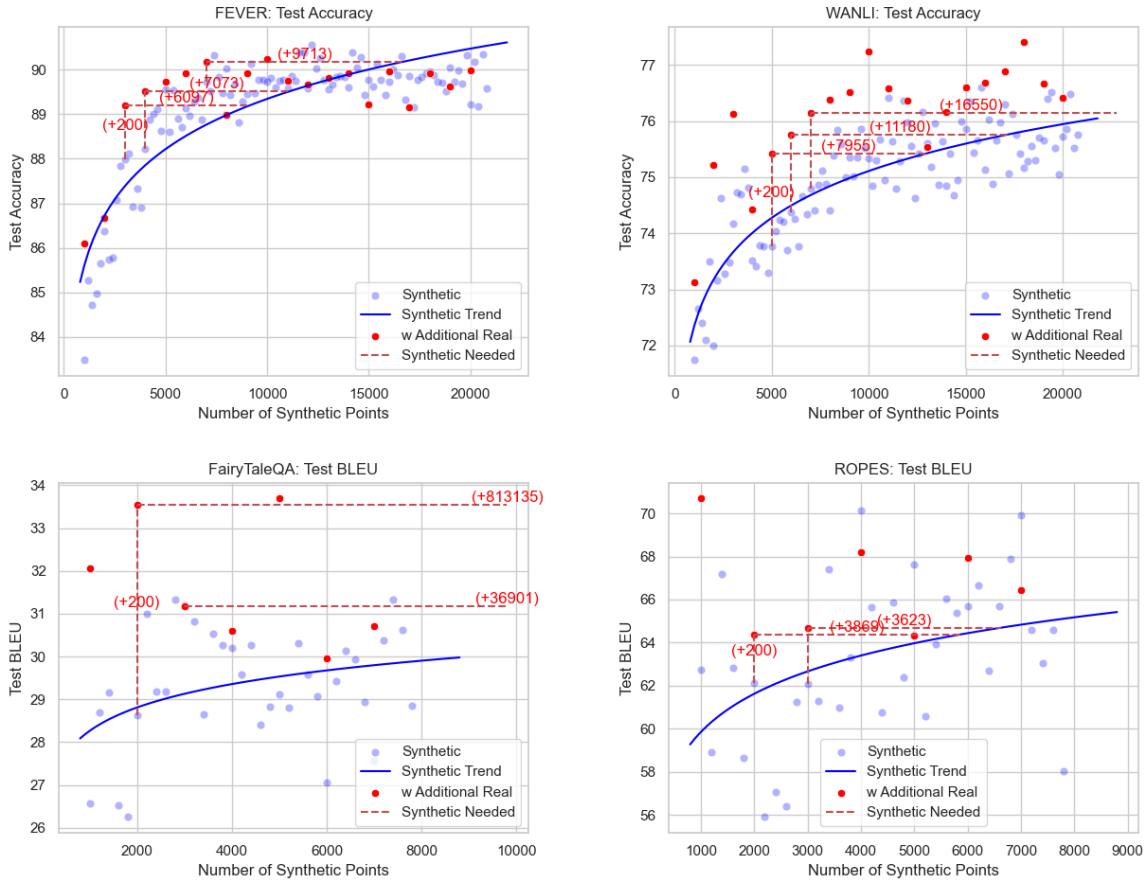


Figure 18: Adding 200 real data points is as effective as adding an order of magnitude more synthetic data points.

Fact Verification Prompt Example

System: Given the context, come up with a **True** claim.
Answer in the format below:
Input: This is the third time in two years that the company has lost its director of legal affairs.
Output: Claim: The company has lost a director of legal affairs twice in two years. | **Label**: True

Input: A study of the health status of the elderly in the United States.
Output: Claim: The health status of the elderly can be studied. | **Label**: True
Input: For more than a decade, the town has had a curfew for children, who are to be in bed by 10 p.m.

Output: [MODEL OUTPUT] <- Synthetic Claim

Question Answering Prompt Example

System: Given the context, come up with a question and answer pair. Answer in the format below:
Input: A large log placed in a fire will burn relatively slowly. If the same mass of wood were added to the fire in the form of small twigs, they would burn much more quickly. ....Group S made a fire using large logs, while group F made a fire using small twigs and sawdust. Both groups used the same volume of wood.
Output: Question: Which group made a fire from more particles of wood? | **Answer**: group F
Input: [NEW CONTEXT]

Ouput: [MODEL OUTPUT] <- Synthetic Question + Ans

Figure 19: Example prompts used to synthetically generate (claim, label) or (question, answer) pairs using a new context / evidence text.

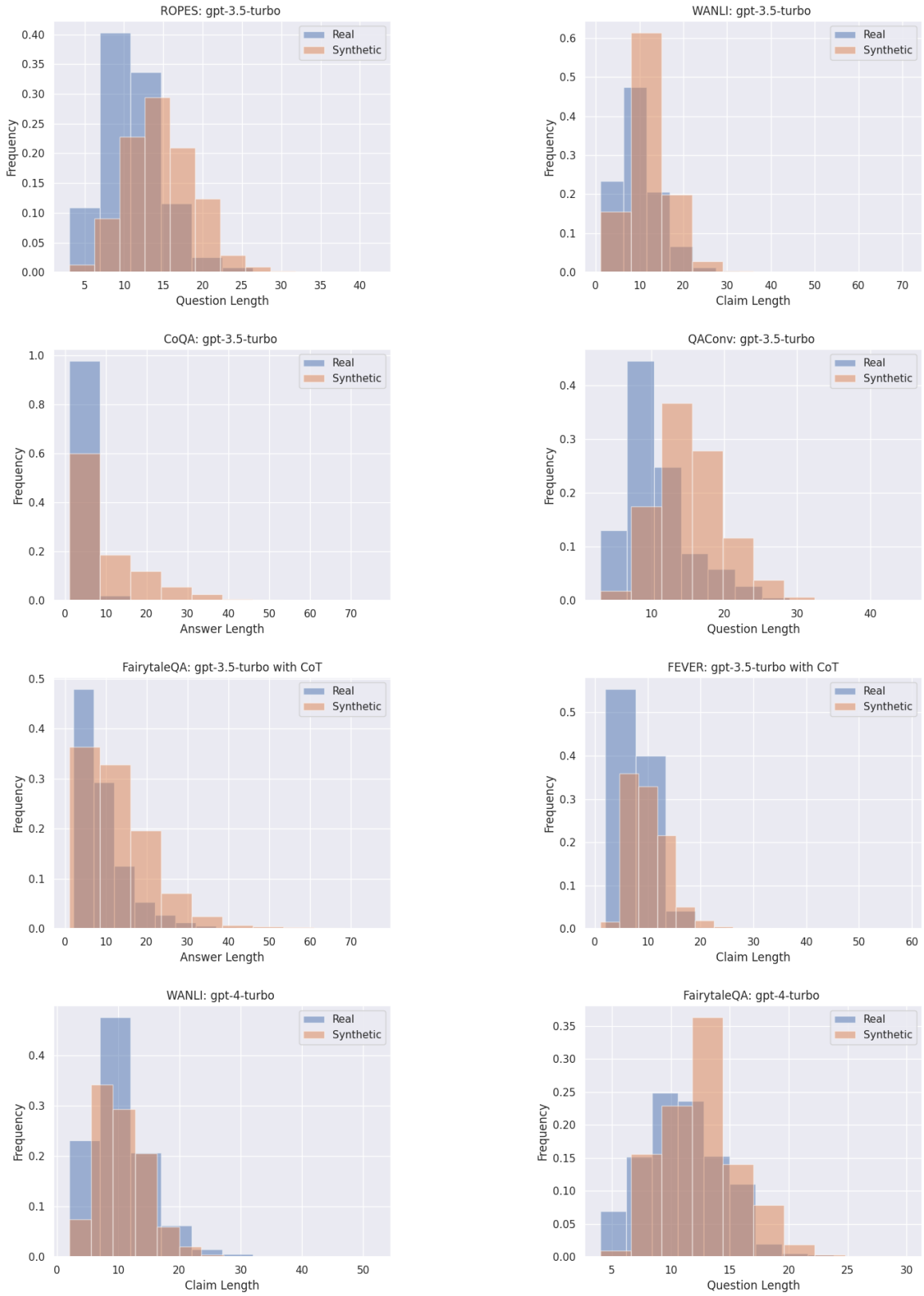


Figure 20: Synthetic data is, on average, longer than its human generated counterpart. This trend can be seen on FV (claims) and QA (claims and questions), and holds across prompt models and strategies.

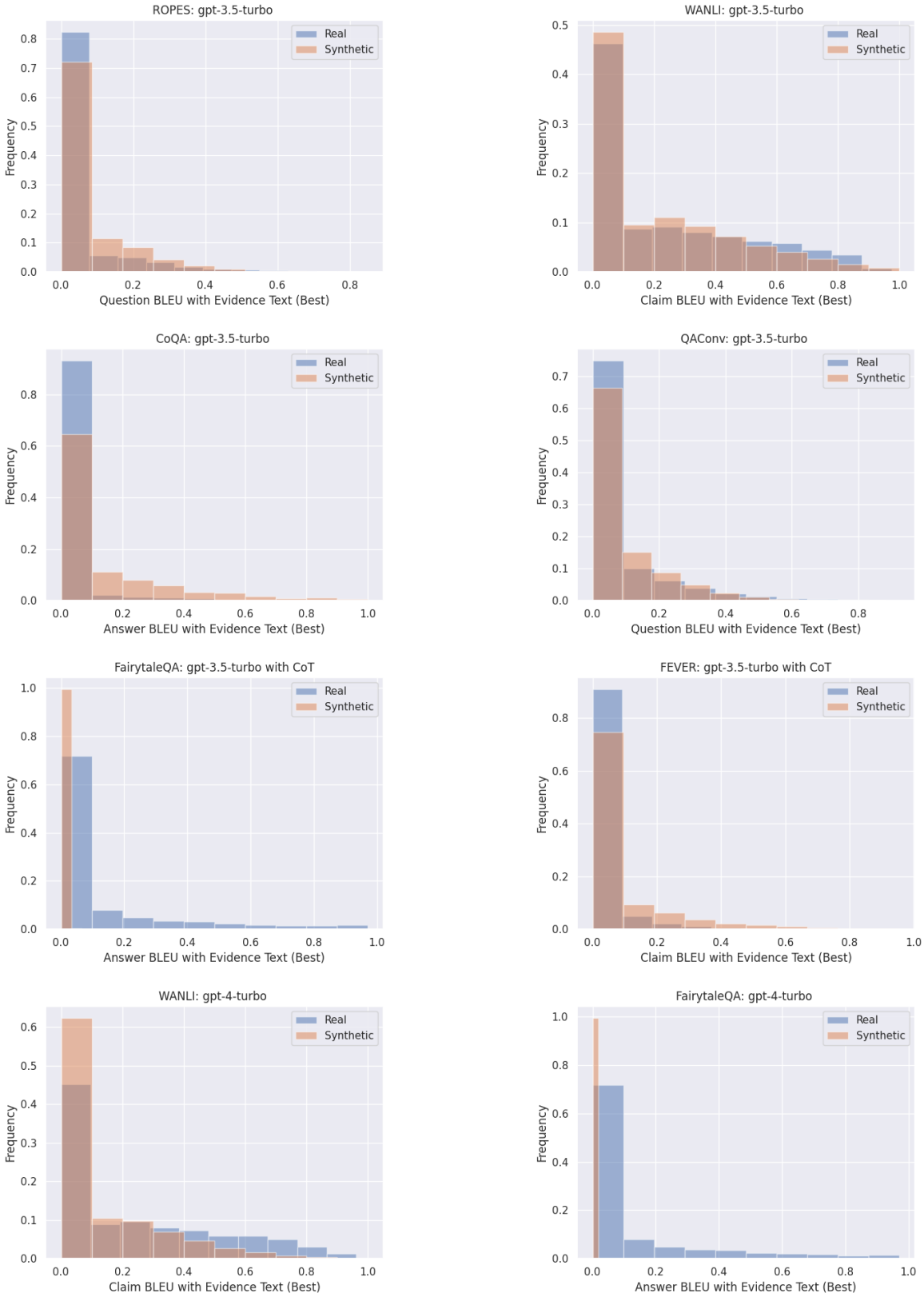


Figure 21: Synthetic data generally exhibits a higher maximum BLEU score measured against sentences from the context. This suggests that synthetic questions, answers, and claims are more extractive than their human generated counterparts

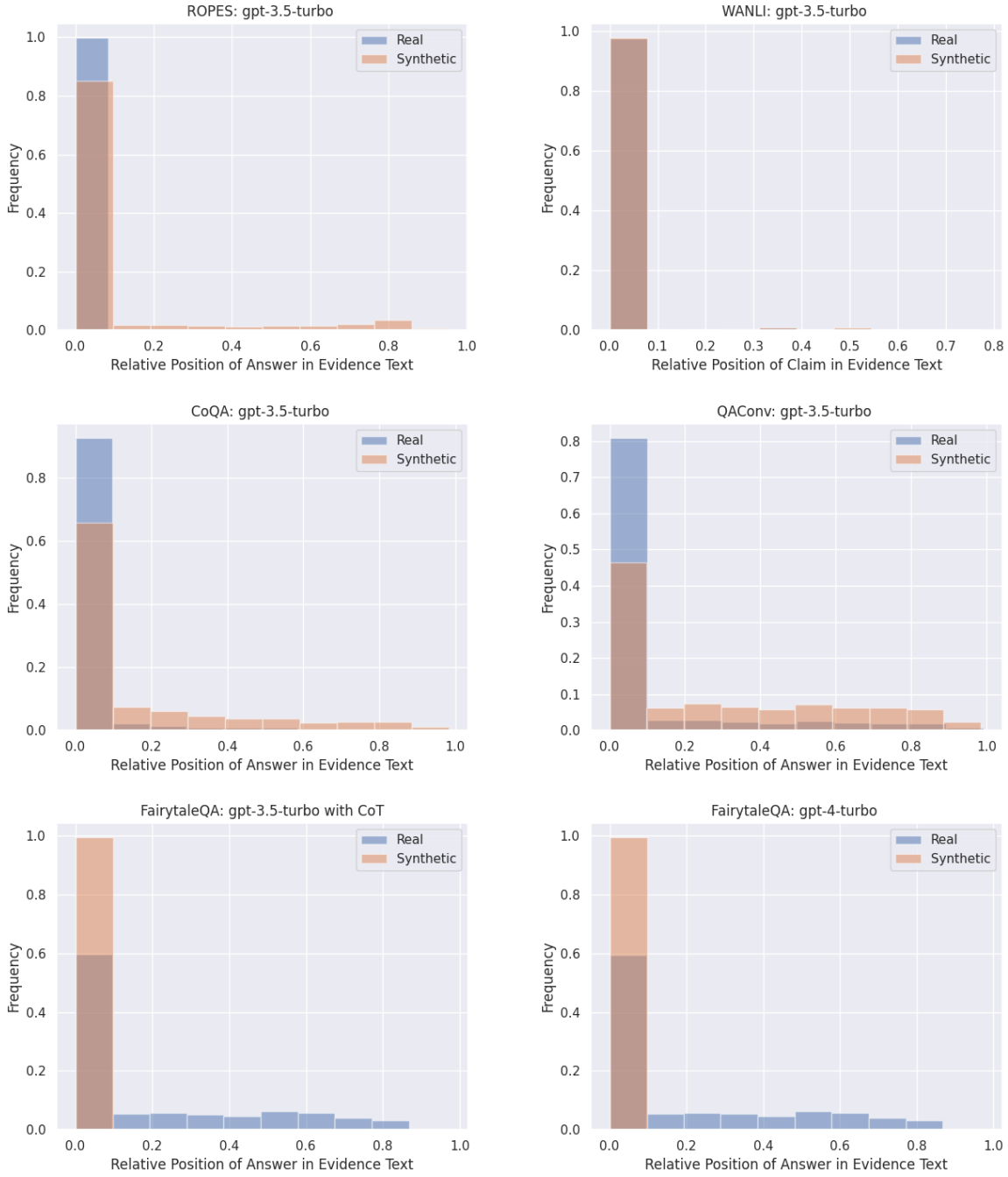


Figure 22: Synthetic data typically chooses more diverse sources (in terms of answer location or claim location in the evidence text), while humans tend to favor the start of the evidence text.

Seeking Rational Demonstrations for Large Language Models: A Domain Generalization Approach to Unsupervised Cross-Domain Keyphrase Generation

Guangzhen Zhao†, **Yu Yao**‡, **Dechang Kong**†, **Zhenjiang Dong**†*
† School of Computer Science, Nanjing University of Posts and Telecommunications, China
‡ School of Computer Science and Artifical Intelligence, Chaohu University, China
{zhaogz, dongzhenjiang, b21030713}@njupt.edu.cn, 054096@chu.edu.cn

Abstract

Unsupervised cross-domain keyphrase generation (KPG) is crucial in real-world natural language processing scenarios. However, the accuracy of up-to-date approaches is limited by the distribution shift between source and target domain, which stems from the cross-domain field. Large language models (LLMs) offer potential for the cross-domain keyphrase generation tasks due to their strong generalization abilities, facilitated by providing demonstrations relevant to the target task. Nevertheless, it is often difficult to obtain labeled samples from the target domain. To address this challenge, this paper aims to seek rational demonstrations from the source domain, thereby improving the LLMs' ability in the unsupervised cross-domain keyphrase generation setting. Specifically, we design a novel domain-aware retrieval model on the source domain. Guided by insights from domain generalization theory, we introduce two generalization terms, one for cross-domain relevance and another for each domain consistency to better support retrieval of rational demonstrations. By the retrieved source-domain demonstrations and distance-based relevant score, the proposed approach achieves optimal accuracy. Comprehensive experiments on widely used cross-domain KPG benchmarks demonstrate our approach's state-of-the-art performance and effectiveness.

1 Introduction

Keyphrase Generation (KPG) is critical in identifying discriminative information (Meng et al., 2017; Shao et al., 2024; Boudin and Aizawa, 2024). As a fundamental natural language generation (NLG) task, keyphrase generation facilitates a wide variety of downstream applications, including document clustering (Hulth and Megyesi, 2006; Chiu et al., 2020), information retrieval (Ushiku et al., 2017; Boudin et al., 2020) and text summarization (Wang and Cardie, 2013). However, in real world scenarios, e.g. news articles (Gallina et al., 2019), web pages (Xiong et al., 2019), technical question-answer website (Yuan et al., 2020), obtaining labels can be challenging due to the reliance on domain expertise, or even infeasible because of the strict privacy constraints. Deploying keyphrase generation model in these scenarios often occurs in cross-domain KPG setting, e,g., first training KPG models on source tasks and then generalizing to target tasks (Gulrajani and Lopez-Paz; Wiles et al.). What is notable is that distribution shift is an inevitable phenomenon in cross-domain KPG.

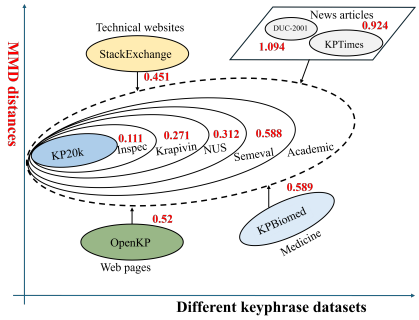


Figure 1: we treat the KP20k train dataset as the source domain, and using MMD to measure the degree of the distribution shift between different datasets.

As depicted in Figure 1, we treat the KP20k train dataset (Meng et al., 2017) as the source domain, and adopt Maximum Mean Discrepancy (MMD) (Kim et al., 2016) to measure the degree of the distribution shift. Higher MMD values between each two datasets indicate increased requirements for the model's generalization capabilities. For example, compared to academic domain datasets, such as Inspec (Hulth, 2003), Krapivin (Krapivin and Marchese, 2009), and NUS (Nguyen and Kan, 2007), the MMD distance is significantly larger for other domain datasets (e.g., OpenKP (Xiong et al., 2019), KPBiomed (Houbre et al., 2022)). This makes generalizing the KPG model in cross-domain setting more challenging than in standard generation,

* Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 414–424
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

where a robust decision boundary is sufficient. An effective strategy to alleviate distribution shift is the use of large language models (LLMs), where labeled samples are used as prompts to facilitate the generation of target domain keyphrases. However, even annotating a limited number of samples can be prohibitively expensive, and often impractical due to the need for expert annotators (Chau et al., 2020; Boudin and Aizawa, 2024). How to effectively leverage existing labeled source domain samples to prompt LLMs in unsupervised cross-domain KPG is a crucial problem that deserves in-depth exploration. In this work, we propose a seeking rational demonstration (SRD) approach for enhancing LLMs' ability in unsupervised cross-domain KPG. The SRD approach is designed based on DPR (Dense Passage Retrieval) model (Karpukhin et al., 2020). Specifically, we randomly select 20% samples from KP20k train datasets and target test dataset as the query set, with the rest as candidate set. We then construct positive and negative samples by computing the relevance of each query-candidate sample pair. However, there is no guarantee of accuracy when mapping the target query samples to the rational KP20k train samples during test time when facing distribution shift. As such, we introduce the domain projection simulation (DPS) with two new losses based on domain generalization theory to alleviate distribution shift between training and test phrases. The first loss is a domain projection loss based on the MMD distance, which helps learn more domain-invariant features and mitigates distribution shift. The second one is a domain characteristic loss that encourages the learnable train distribution to be orthogonal in the representation space, thus increasing the diversity and preventing domain characteristic vanished.

Our contributions are three-fold:

- We quantify the distribution shift phenomenon when applying LLMs to unsupervised cross-domain KPG. To the best of our knowledge, this is the first work that introduces domain generalization with LLMs in KPG.
- We investigate both theoretically and empirically how domain generalization technique can help with distribution projection, and thus alleviating distribution shift.
- Experimental results demonstrate that our SRD approach performs better than up-to-date baselines on widely used cross-domain KPG test datasets. The processed test datasets, codes and experimental results will be upload at https://github.com/chrischowfy/SRD/.

2 Realted Work

Current keyphrase generation tasks primarily focus on academic article domain, encompassing both supervision (Meng et al., 2017; Ye et al., 2021; Shao et al., 2024; Kang and Shin, 2024) and unsupervison (Shen et al., 2022; Boudin and Aizawa, 2024) fashions. AutoKeyGen (Shen et al., 2022) is the first work to explore the unsupervised keyphrase generation task, which trains a sequence-to-sequence model to extract present keyphrases and synthesize absent keyphrases using a phrase corpus. After that, UOKG (Do et al., 2023) extends AutoKeyGen in open-domain setting, which generates keyphrases that represent the core concept of the source text. Another related work is domain-adaptive keyphrase generation (Meng et al., 2023), however, it requires a few labeled samples from the target domain for fine-tuning. Another related direction is keyphrase generation in low-resource settings (Wu et al., 2022; Garg et al., 2023). Existing researches often employ target domain data pretraining (Wu et al., 2022) and data augmentation (Garg et al., 2023) to enhance keyphrases generation capabilities in low-resource contexts. However, in the era of large language models, considering the challenges of fine-tuning LLMs and computational expenses, there is an urgent need for new research approaches to address the problem of unsupervised cross-domain keyphrase generation.

Recently, LLMs perform well in unsupervised keyphrase generation (MartÍłnez-Cruz et al., 2024). Notably, when provided with an in-context prompt that contains a few relevant samples from the target domain, LLMs can further enhance their keyphrase generation performance in target domain (Jiang et al., 2024). However, in real-world scenarios, acquiring labeled target samples is non-trivial. This motivates our work: how to effectively prompt LLMs using labeled source domain samples to improve keyphrase generation across different domains.

3 Methodology

3.1 Problem Settings

Formally, the unsupervised cross-domain KPG task is a text generation problem conditioned on both

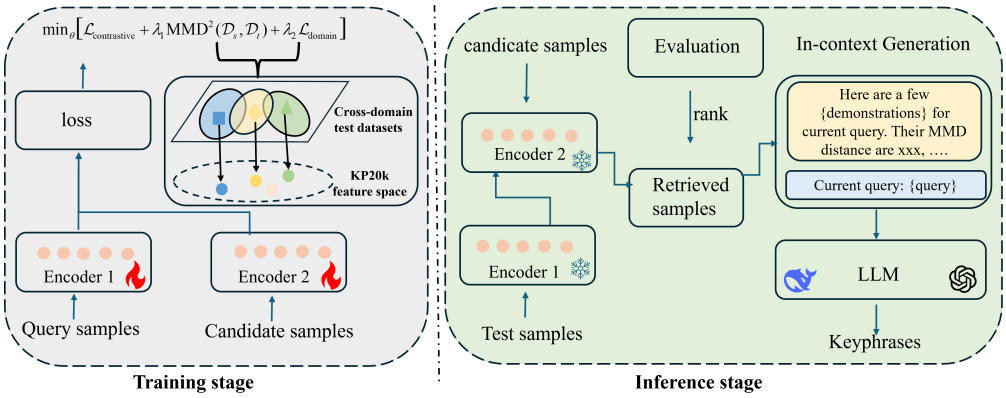


Figure 2: Overview of the seeking rational demonstrations (SRD) approach for unsupervised cross-domain KPG.

the target domain input and a set of source domain demonstrations:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{y} \mid x_t, \text{Retrieve}(x_t, \mathcal{S}), \Theta), \quad (1)$$

where $x_t \in \mathcal{D}_t$ is the unlabeled target-domain input, $\text{Retrieve}(\cdot,\cdot)$ denotes a retrieval function that selects relevant demonstrations from the labeled source-domain samples $\mathcal{S}$, and Θ denotes the LLMs' parameters. Note that no labeled target-domain data is available, the generalization capacity of the model is crucially depends on the quality of its retrieved source-domain samples under distribution shift.

3.2 Basic Retrieve Module

As shown in Figure 2, to efficiently retrieve rational demonstrations for unsupervised cross-domain KPG, we first build a basic dual-encoder retrieval module based on the DPR model (Karpukhin et al., 2020). The module includes two encoders, which can map both query text and candidate text into vector representations.

Constructing Positive & Negative Pairs. Since the KP20k train dataset does not explicitly provide positive or negative sample pairs, we manually construct them by analyzing the keyphrases of each sample. Specifically, let $\mathbf{k}_x$ and $\mathbf{k}_s$ be the keyphrases of samples x and s, respectively. The relevance score is defined as:

$$\text{Rel}(x,s) = \alpha \cdot \text{sim}_{\text{embed}}(\mathbf{k}_x, \mathbf{k}_s) + (1-\alpha) \cdot \text{Jaccard}(\mathbf{k}_x, \mathbf{k}_s), \quad (2)$$

where $\text{sim}_{\text{embed}}(\mathbf{k}_x, \mathbf{k}_s)$ denotes the semantic similarity and $\text{Jaccard}(\mathbf{k}_x, \mathbf{k}_s)$ is the Jaccard similarity. If $\text{Rel}(x,s)$ exceeds a global threshold γ_1, we classify (x,s) as a positive (matched) pair, and otherwise treat it as a negative (unmatched) pair.

Self-supervised Demonstrations Evaluator. We design a self-supervised evaluator utilizing the KP20k train dataset to assess the efficacy of retrieved demonstrations. We divide KP20k into two disjoint subsets: the pseudo-source domain $\mathcal{D}_{\text{ps}}$ and pseudo-target domain $\mathcal{D}_{\text{pt}}$. The former is used to simulate the source domain for retrieval purposes, and the latter is used to mimic the unlabeled target domain scenario. We then train a regression model to learn a function mapping the triplet $(\mathbf{x}, \mathbf{e}, \mathbf{y}^{\text{gen}})$ to a quality score $\hat{s}$, where $\mathbf{x}$ represents the abstract text in the pseudo-target domain, $\mathbf{e}$ indicates the retrieved examples, and $\mathbf{y}^{\text{gen}}$ are the keyphrases generated by a LLM. Let $\mathbf{y}^{\text{ref}}$ denote the ground-truth keyphrases for $\mathbf{x}$ in the pseudo-target domain. The quality score s of $\mathbf{y}^{\text{gen}}$ is computed by the Eq.2, i.e., $s = \text{Rel}(\mathbf{y}^{\text{gen}}, \mathbf{y}^{\text{ref}})$. More details can be found in Appendix A.1 Implementation Details.

3.3 The Risk of Distribution Shift

We first give the distribution shift measured by $\mathcal{H}$-divergence (Ben-David et al., 2010):

$$d_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) = 2 \sup_{h \in \mathcal{H}} \left| \Pr_{x \sim \mathcal{D}_s}[h(x)=1] - \Pr_{x \sim \mathcal{D}_t}[h(x)=1] \right|, \quad (3)$$

where classifier $h : \mathcal{X} \to \{0,1\}$ is a labeling function on each domain sample. Next, the ideal target domain $\bar{\mathcal{D}}_t$ is assumed that lies in the source domain convex hull Λ_s (Albuquerque et al., 2019). Under this assumption, the risk $\epsilon_t(h)$ on the target domain $\mathcal{D}_t$ is upper-bounded (Albuquerque et al., 2019; Chen et al., 2019) by:

$$\epsilon_t(h) \leq \sum_{i=1}^{K} \eta_i \epsilon_s^i(h) + \gamma + \zeta. \quad (4)$$

The first term is the risks over source domains, which can be minimized by empirical risk minimization. The second term γ is the $\mathcal{H}$-divergence

between the ideal target $\overline{\mathcal{D}}_t$ and the real target domain D_t. The third term could be disregarded as it is negligibly small. In this paper, we focus on the first and second terms to alleviate distribution shift in unsupervised cross-domain KPG.

3.4 Distribution Projection Simulation

Since there is distribution shift existed between source and target domain, we attempt to mitigate the ensuing cross-domain risk by domain generalization theory. Specifically, we employ MMD (Kim et al., 2016) as a kernel-based criterion that directly measures the similarly between the source and target samples in its representation space. By guiding the encoder to produce more domain-invariant features, we reduce adverse effects of distribution shift and thus improve generalization in unsupervised cross-domain KPG. In each training iteration, we select a mini-batch of labeled source domain samples $\{x_i^s, y_i^s\}_{i=1}^m \subset \mathcal{D}_s$ and unlabeled target domain samples $\{x_j^t\}_{j=1}^n \subset \mathcal{D}_t$. Using the kernel function $k(\cdot,\cdot)$, the squared MMD is computed as:

$$\mathrm{MMD}^2(\mathcal{D}_s, \mathcal{D}_t) = \frac{1}{m(m-1)} \sum_{\substack{i,j=1 \\ i\neq j}}^{m} k(f(x_i^s), f(x_j^s)) + \frac{1}{n(n-1)} \sum_{\substack{i,j=1 \\ i\neq j}}^{n} k(f(x_i^t), f(x_j^t)) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(f(x_i^s), f(x_j^t)), \tag{5}$$

where $f(\cdot)$ is the encoder's mapping from input text to a feature vector, $k(\cdot,\cdot)$ is the radial basis function kernel with $k_{i,j} = k(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|)$, γ is a hyperparameter (Kim et al., 2016). MMD imposes a domain-invariance constraint on the encoders' representations. In this way, the features between the source and target domain samples align more closely, which are crucial for accurate keyphrase generation under distribution shift. However, a naive reduction of MMD operates on mean embedding feature of each sample, which may be insufficient to characterize sample mapping (Vayer and Gribonval, 2023). Hence, we also incorporate domain-specific features into the encoders that retain critical domain cues.

Preserving Domain Characteristics via Orthogonality. To further impose domain features into the encoders, we introduce two sets of mean and variance vectors:$\{\mu_b^s, \sigma_b^s\}$ for the source domain samples, and$\{\mu_b^t, \sigma_b^t\}$ for the target domains samples. We design a domain (orthogonality) loss to enforce these vectors to remain domain-specific:

$$\mathcal{L}_{\text{domain}} = \left| \frac{\mu_b^s}{\|\mu_b^s\|} \cdot \frac{\mu_b^t}{\|\mu_b^t\|} \right| + \left| \frac{\sigma_b^s}{\|\sigma_b^s\|} \cdot \frac{\sigma_b^t}{\|\sigma_b^t\|} \right|, \tag{6}$$

where $\|\cdot\|$ is the vector norm, and $\cdot$ is the dot product. By forcing the normalized means and variances from the source and target to be approximately orthogonal, we ensure that while we reduce overall distribution divergence, each domain maintains its own "style" (i.e., domain-specific signals).

3.5 Training

We train the dual-encoder retriever by combining traditional contrastive loss (Karpukhin et al., 2020) with the standard MMD alignment. The overall objective is to minimize:

$$\min_{\theta} \left[\mathcal{L}_{\text{contrastive}} + \lambda_1 \,\mathrm{MMD}^2(\mathcal{D}_s, \mathcal{D}_t) + \lambda_2 \,\mathcal{L}_{\text{domain}} \right], \tag{7}$$

where θ is the parameter of the retriever, and λ_1 and λ_2 are hyperparameters balancing the three terms. By jointly minimizing the contrastive loss and domain relevant loss, we reduce the distribution shift (thus addressing the second risk term in Eq. (4)) while preserving crucial domain characteristics in both source and target domains.

4 Experiments

4.1 Experimental Setup

Datasets We conduct experiments on five public domain KPG datasets, namely DUC-2001, KPTimes, OpenKP, StackExchange and KPBiomed. The detailed statistic is depicted in Table 1.

Datasets	Type	Test docs	#Kps/doc	%Absent	Api_llm
StackExchange (Yuan et al., 2020)	Technology	2000	2.65	48.78	300
DUC-2001 (Wan and Xiao, 2008)	News	308	8.06	2.7	300
KPTimes (Gallina et al., 2020)	News	2000	5.07	54.2	300
OpenKP (Xiong et al., 2019)	Web	2000	2.31	11.62	300
KPBiomed (Houbre et al., 2022)	Medicine	2000	5.38	38.74	300

Table 1: Statistic of cross-domain test datasets.

Baselines and Metrics We compare our approach with AutoKeyGen (Shen et al., 2022) and UOKG (Do et al., 2023), which are the only two standard baselines on unsupervised cross-domain keyphrase generation. Furthermore, we re-implement CopyRNN (Meng et al., 2017), One2set (Ye et al., 2021) and DeepKPG (Wu et al., 2023) as supervised baselines. The subscript in Table 2 indicates the standard deviation (e.g., 9.4_7 denotes 9.40±0.7).We also adopt several LLM backbones to evaluate our approach. Following UOKG, we use the F1@K and Recall@K as evaluation metrics, where K is the number of predicted keyphrases to be considered. We only use KP20k to seek rational demonstrations.

Present keyphrase generation											
Methods	StackExchange		DUC-2001		KPTimes		OpenKP		KPBiomed		Average
	F1@3	F1@5	F1@3	F1@5	F1@3	F1@5	F1@3	F1@5	F1@3	F1@5	
AutoKeyGen	13.7_6	14.6_2	6.6_5	9.1_1	11.1_2	11.5_1	8.9_5	9.1_2	12.9_1	15.2_4	13.70
UOKG	16.2_1	$\underline{18.3}_4$	16.0_7	18.7_1	12.2_7	13.1_7	13.3_5	11.7_7	14.4_3	14.6_4	14.85
CopyRNN	$\mathbf{24.9}_2$	$\mathbf{22.6}_3$	7.3_1	8.7_2	17.3_5	17.4_4	10.9_2	10.3_1	21.7_3	22.8_2	16.39
One2Set	18.2_3	15.4_4	7.9_2	9.1_7	10.8_2	6.4_3	6.5_3	6.1_6	20.7_7	20.9_2	12.20
DeepKPG	16.3_1	11.4_6	13.1_2	11.1_3	15.7_5	11.5_3	16.1_7	11.4_2	30.8_4	24.7_5	16.21
Qwen2.5-14b	1.8_1	1.2_7	8.4_5	6.8_3	4.5_2	3.1_6	7.6_1	5.4_5	5.4_4	4.1_2	4.83
Llama3.3-70b	5.0_6	3.4_4	21.3_5	19.1_5	13.1_4	9.8_7	19.9_7	14.5_2	20.9_6	19.4_2	14.64
ChatGPT-3.5	5.4_4	4.1_1	21.5_7	21.7_3	12.5_4	9.3_1	19.1_3	13.7_6	18.0_2	16.8_5	14.21
GPT4o	9.9_1	7.2_1	26.6_4	27.2_3	20.1_1	16.4_2	23.9_3	17.2_4	29.3_2	26.1_3	20.39
DeepSeek	5.4_3	3.7_4	21.6_3	21.9_5	11.0_2	8.5_3	16.5_2	12.2_1	18.4_2	17.2_3	13.64
Ours(Qwen2.5-14b)	17.8_3	14.3_7	25.2_3	26.1_5	14.2_6	10.4_5	22.6_6	20.1_2	25.8_7	24.6_2	20.11
Ours(ChatGPT-3.5)	16.8_2	12.1_5	24.5_1	25.1_2	16.9_5	14.6_6	24.7_4	18.6_6	27.4_1	25.1_7	20.58
Ours(GPT4o)	$\underline{23.1}_4$	16.1_3	$\underline{27.2}_2$	$\underline{28.8}_4$	$\mathbf{28.9}_1$	$\mathbf{23.6}_1$	$\mathbf{29.2}_4$	$\mathbf{22.4}_2$	$\mathbf{33.0}_3$	$\mathbf{32.7}_3$	**26.50**
Ours(DeepSeek)	21.2_2	14.9_4	$\mathbf{29.1}_1$	$\mathbf{29.0}_1$	$\underline{27.7}_2$	$\underline{22.7}_3$	$\underline{27.1}_4$	$\underline{19.6}_4$	$\underline{31.3}_1$	$\underline{30.0}_3$	$\underline{25.26}$
Absent keyphrase generation											
Methods	R5	R10	R5	R10	R5	R10	R5	R10	R5	R10	Average
AutoKeyGen	0.9_5	1.0_2	–	–	0.2_1	0.2_1	0.1_3	0.3_1	0.4_2	0.8_2	0.49
UOKG	1.9_2	2.8_5	–	–	1.1_4	1.1_3	0.3_2	0.5_3	1.4_2	2.2_5	1.41
CopyRNN	2.5_2	3.7_4	–	–	0.6_4	0.8_1	0.1_4	0.1_4	0.8_3	1.2_3	1.22
One2Set	1.1_1	1.5_5	–	–	0.3_5	0.3_1	0.1_5	0.2_3	0.8_2	0.8_5	0.64
DeepKPG	2.2_1	2.3_1	–	–	1.3_5	1.4_5	2.3_1	2.4_2	2.9_4	3.0_3	2.23
Qwen2.5-14b	0.2_2	0.2_1	–	–	0.5_2	0.7_4	3.1_3	3.0_3	0.7_4	0.7_5	1.14
Llama3.3-70b	2.9_4	3.2_1	–	–	2.3_1	2.3_1	7.6_3	7.8_2	3.2_2	3.5_1	4.10
ChatGPT-3.5	2.3_2	3.0_4	–	–	4.1_6	4.2_4	9.9_6	9.4_4	1.8_2	1.8_6	4.56
GPT4o	6.5_3	7.5_4	–	–	7.7_1	8.1_2	9.8_2	10.6_3	4.2_1	4.3_3	7.34
DeepSeek	3.1_1	3.2_3	–	–	5.8_4	6.0_2	9.8_4	9.7_1	2.9_1	2.9_2	5.42
Ours(Qwen2.5-14b)	6.7_2	6.3_4	–	–	2.1_4	1.9_3	8.4_2	8.9_3	4.0_3	3.9_4	5.28
Ours(ChatGPT-3.5)	4.9_4	5.2_1	–	–	6.8_2	7.2_2	9.9_2	12.4_3	3.9_2	4.7_2	6.88
Ours(GPT4o)	$\mathbf{8.8}_4$	$\underline{9.1}_4$	–	–	$\mathbf{12.3}_4$	$\mathbf{13.4}_3$	$\underline{12.1}_1$	$\underline{12.7}_5$	$\mathbf{5.3}_3$	$\mathbf{5.8}_5$	**9.94**
Ours (DeepSeek)	$\underline{8.7}_1$	$\mathbf{9.4}_3$	–	–	$\underline{10.9}_2$	$\underline{11.1}_2$	$\mathbf{12.3}_1$	$\mathbf{12.8}_3$	$\underline{4.6}_3$	$\underline{4.8}_2$	$\underline{9.33}$

Table 2: Performances of cross-domain test datasets (%).

4.2 Comparisons to the State-of-the-art Methods

Table 2 shows the performance of different methods on the five cross-domain test datasets, the proposed SRD method consistently performs better than the SOTA methods across multiple datasets. Ours(GPT4o) method exceeds ChatGPT-3.5 by 12.29% and 5.38% in absolute average values on present and absent keyphrase generations. The metrics of ChatGPT-3.5 and Ours (ChatGPT-3.5) (14.21 vs. 20.58) indicate reasonable demonstrations significantly enhance the accuracy in unsupervised cross-domain KPG. Furthermore, we also find that LLMs tend to weaken their generation capabilities on free-form text, while reasonable demonstrations effectively mitigate this negative impact. Besides, we can see that the DeepSeek model is competitive with GPT4o (26.50/9.94 vs. 25.26/9.33) on present and absent keyphrase, indicating the high cost-effectiveness of the former.

4.3 Sampling Ratio Analysis

Figure 3 shows the performance of various proportions of sample selections: 5%, 10%, 20%, 30%, and 40%. It is observed that a lower proportion of the query set results in insufficient training diversity, thereby constraining the overall effectiveness of KPG. As the proportion increases, the performance of KPG improves accordingly, reaching its peak around the 30% inclusion mark. This suggests that the optimal query set proportion likely lies between 20% and 30%. However, when the proportion is increased to 40%, the query set tends to accumulate redundant or highly similar samples.

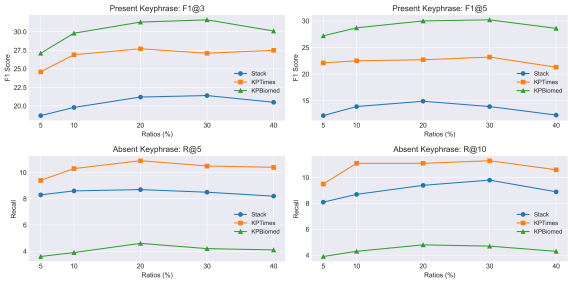


Figure 3: Performances under different ratios (%).

5 Conclusion

In this work, we present a new perspective for unsupervised cross-domain KPG with prompts LLMs to enhance its generalizability. We propose a novel seeking rational demonstrations (SRD) approach, which is the first work that introduces domain generalization on retrieve model for unsupervised cross-domain KPG. We design MMD driven distribution project loss and orthogonality loss beyond the mapping of positive and negative samples. Extensive experiments demonstrate the better performance and the effectiveness of the proposed seeking rational demonstrations approach.

6 Limitations

In addressing the practical distribution shift challenges faced by large language models in unsupervised cross-domain PKG, i.e., the difficulty of obtaining labeled target domain samples for prompting, this paper proposes a seeking rational demonstration approach. Using the Maximum Mean Discrepancy (MMD) distance as a metric for disparity, this approach retrieves highly relevant samples from the source domain to serve as demonstrations for the target samples. The main limitations of this paper are as follows: 1. The setup of the evaluator is rather rudimentary. Due to the unavailability of labeled samples from the target domain, only source domain samples are used. When encountering domains that significantly differ from the source domain, this may impact the selection of demonstrations and thus affect the quality of unsupervised cross-domain keyphrase generation. Future work could explore joint optimization of retriever and evaluator. 2. This approach does not involve optimizations for domain generalization of the large language model itself. If domain-specific instruction fine-tuning could be applied to the language model, it is believed that the model performance could be further enhanced.

7 Acknowledgements

The authors thank the anonymous reviewers for their valuable comments. This work is supported by the Scientific Research Fund of Nanjing University of Posts and Telecommunications under project numbers XK0040924146 and XK0040925024, and in part by the Scientific Research Fund of Chaohu University under project number KYQD-2025029.

References

Isabela Albuquerque, João Monteiro, Mohammad Darvishi, Tiago H Falk, and Ioannis Mitliagkas. 2019. Generalizing to unseen domains via distribution matching. *arXiv preprint arXiv:1911.00804*.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*, 79:151–175.

Florian Boudin and Akiko Aizawa. 2024. Unsupervised domain adaptation for keyphrase generation using citation contexts. In *Findings of the Association for Computational Linguistics: EMNLP 2024 (Findings of EMNLP)*, pages 598–614, Miami, Florida, USA.

Florian Boudin, Ygor Gallina, and Akiko Aizawa. 2020. Keyphrase generation for scientific document retrieval. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, (ACL-20)*, pages 1118–1126, Online.

Hung Chau, Saeid Balaneshin, Kai Liu, and Ondrej Linda. 2020. Understanding the tradeoff between cost and quality of expert annotations for keyphrase extraction. In *Proceedings of the 14th Linguistic Annotation Workshop*, pages 74–86, Barcelona, Spain.

Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. 2019. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International Conference on Machine Learning, (ICLR-19)*.

Billy Chiu, Sunil Kumar Sahu, Derek Thomas, Neha Sengupta, and Mohammady Mahdy. 2020. Autoencoding keyword correlation graph for document clustering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-20)*, pages 3974–3981, Online.

Lam Do, Pritom Saha Akash, and Kevin Chen-Chuan Chang. 2023. Unsupervised open-domain keyphrase generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10614–10627, Toronto, Canada.

Ygor Gallina, Florian Boudin, and Beatrice Daille. 2019. KPTimes: A large-scale dataset for keyphrase generation on news documents. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 130–135, Tokyo, Japan.

Ygor Gallina, Florian Boudin, and Béatrice Daille. 2020. Large-scale evaluation of keyphrase extraction models. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, JCDL '20, pages 271–278, New York, NY, USA. Association for Computing Machinery.

Krishna Garg, Jishnu Ray Chowdhury, and Cornelia Caragea. 2023. Data augmentation for low-resource keyphrase generation. In *Findings of the Association for Computational Linguistics: ACL 2023 (Findings of ACL)*, pages 8442–8455, Toronto, Canada.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems*. MIT Press.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *Proceedings of the International Conference on Learning Representations, (ICLR-20)*.

Maël Houbre, Florian Boudin, and Beatrice Daille. 2022. A large-scale dataset for biomedical keyphrase generation. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, pages 47–53, Abu Dhabi, United Arab Emirates (Hybrid).

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, (EMNLP-03)*, pages 216–223.

Anette Hulth and Beáta B. Megyesi. 2006. A study on automatically extracted keywords in text categorization. In *Proceedings of the International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics, (COLING/ACL-06)*, pages 537–544, Sydney, Australia.

Zhongtao Jiang, Yuanzhe Zhang, Kun Luo, Xiaowei Yuan, Jun Zhao, and Kang Liu. 2024. On the in-context generation of language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10169–10187, Miami, Florida, USA.

Byungha Kang and Youhyun Shin. 2024. Improving low-resource keyphrase generation through unsupervised title phrase generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8853–8865.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online.

Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. 2016. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems*, volume 29.

M. Krapivin and M. Marchese. 2009. Large dataset for keyphrase extraction. *Technical report, University of Trento*.

R. MartĺłNez-Cruz, A.J. LĺőPez-LĺőPez, and Portela. 2024. Chatgpt vs state-of-the-art models: a benchmarking study in keyphrase generation task. *Appl. Intell.*, 55(50).

Rui Meng, Tong Wang, Xingdi Yuan, Yingbo Zhou, and Daqing He. 2023. General-to-specific transfer labeling for domain adaptable keyphrase generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1602–1618, Toronto, Canada.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-17)*, pages 582–592, Vancouver, Canada.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pages 317–326.

Liangying Shao, Liang Zhang, Minlong Peng, Guoqi Ma, Hao Yue, Mingming Sun, and Jinsong Su. 2024. One2Set + large language model: Best partners for keyphrase generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP-24)*, pages 11140–11153, Miami, Florida, USA.

Xianjie Shen, Yinghan Wang, Rui Meng, and Jingbo Shang. 2022. Unsupervised deep keyphrase generation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-22)*, pages 11303–11311.

Atsushi Ushiku, Shinsuke Mori, Hirotaka Kameko, and Yoshimasa Tsuruoka. 2017. Game state retrieval with keyword queries. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR-17)*, pages 877–880, Shinjuku, Tokyo, Japan.

Titouan Vayer and Rémi Gribonval. 2023. Controlling wasserstein distances by kernel norms with application to compressive statistical learning. *J. Mach. Learn. Res.*, 24(1).

Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence, (AAAI-08)*, AAAI'08, pages 855ĺC–860. AAAI Press.

Lu Wang and Claire Cardie. 2013. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, (ACL-13)*, pages 1395–1405, Sofia, Bulgaria.

Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre-Alvise Rebuffi, Ira Ktena, Krishnamurthy Dj Dvijotham, and Ali Taylan Cemgil. A fine-grained analysis on distribution shift. In *Proceedings of the International Conference on Learning Representations, (ICLR-22)*.

Di Wu, Wasi Ahmad, and Kai-Wei Chang. 2023. Rethinking model selection and decoding for keyphrase generation with pre-trained sequence-to-sequence models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP-23)*, pages 6642–6658, Singapore.

Di Wu, Wasi Ahmad, Sunipa Dev, and Kai-Wei Chang. 2022. Representation learning for resource-constrained keyphrase generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022 (Findings of EMNLP)*, pages 700–716, Abu Dhabi, United Arab Emirates.

Lee Xiong, Chuan Hu, Chenyan Xiong, Daniel Campos, and Arnold Overwijk. 2019. Open domain web keyphrase extraction beyond language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5175–5184, Hong Kong, China.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021. One2Set: Generating diverse keyphrases as a set. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-21)*, pages 4598–4608, Online.

Xingdi Yuan, Tong Wang, Rui Meng, Khushboo Thaker, Peter Brusilovsky, Daqing He, and Adam Trischler. 2020. One size does not fit all: Generating and evaluating variable number of keyphrases. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, (ACL-20)*, pages 7961–7975, Online.

A Appendix

A.1 Implementation Details

Basic Retrieve Module. To efficiently retrieve suitable demonstrations for unsupervised cross-domain keyphrase generation (KPG), we first build a basic dual-encoder retrieval module based on the DPR model[1]. The module consists of two encoders, which can map both query text and candidate text into vector representations.

Constructing Positive & Negative Pairs. Since the KP20k train dataset does not explicitly provide positive or negative sample pairs, we manually construct them by analyzing the keyphrases of each sample in the dataset. Specifically, let $\mathbf{k}_x$ and $\mathbf{k}_s$ be the keyphrase sets of two samples x and s, respectively. We first apply linguistic preprocessing such as case normalization and stemming[2] to reduce word-level discrepancies. We define the Jaccard similarity as:

$$\text{Jaccard}(\mathbf{k}_x, \mathbf{k}_s) = \frac{|\mathbf{k}_x \cap \mathbf{k}_s|}{|\mathbf{k}_x \cup \mathbf{k}_s|}. \tag{8}$$

A potential limitation of the Jaccard similarity is that keyphrases with low or zero lexical overlap may still have high semantic similarity (e.g., "deep learning" vs. "neural networks"). To better capture deep semantic relationships, we additionally employ an embedding-based metric. Each keyphrase is encoded into a vector $\mathbf{e}_{k_x}$ or $\mathbf{e}_{k_s}$ with *gte-large*[3] embedding model. We compute the cosine similarity between $\mathbf{e}_{k_x}$ and $\mathbf{e}_{k_s}$. The relevance score is defined as:

$$\text{Rel}(x, s) = \alpha \cdot \text{sim}_{\text{embed}}(\mathbf{k}_x, \mathbf{k}_s) + (1-\alpha) \cdot \text{Jaccard}(\mathbf{k}_x, \mathbf{k}_s), \tag{9}$$

where $0 \leq \alpha \leq 1$ is a balancing coefficient. If $\text{Rel}(x, s)$ exceeds a global threshold γ_1, we classify (x, s) as a positive (matched) pair, and otherwise treat it as a negative (unmatched) pair.

Self-supervised Demonstrations Evaluator. To assess the efficacy of retrieved demonstrations in enhancing keyphrase generation within an unlabeled target domain, we design a self-supervised evaluator utilizing the KP20k train dataset. We divide KP20k into two disjoint subsets: the pseudo-source domain $\mathcal{D}_{\text{ps}}$ and pseudo-target domain $\mathcal{D}_{\text{pt}}$ with K-means algorithm[4]. The former is used to simulate the source domain for retrieval purposes, and the latter is used to mimic the unlabeled target domain scenario. We then train a regression model with RoBERTa[5] to learn a function mapping the triplet $(\mathbf{x}, \mathbf{e}, \mathbf{y}^{\text{gen}})$ to a quality score $\hat{s}$, where $\mathbf{x}$ represents the abstract text in the pseudo-target domain, $\mathbf{e}$ indicates the retrieved examples, and $\mathbf{y}^{\text{gen}}$ are the keyphrases generated by LLM. Let $\mathbf{y}^{\text{ref}}$ denote the ground-truth keyphrases for $\mathbf{x}$ in the pseudo-target domain. The quality score s of $\mathbf{y}^{\text{gen}}$ is computed by the defined evaluation metric in Eq.9, i.e., $s = \text{Rel}(\mathbf{y}^{\text{gen}}, \mathbf{y}^{\text{ref}})$. Formally, let f_θ represent the evaluator, the training objective is to minimize the mean squared error (MSE) between $\hat{s}$ and s:

$$\min_\theta \mathbb{E}_{(\mathbf{x}, \mathbf{e}, \mathbf{y}^{\text{gen}}, \mathbf{y}^{\text{ref}}) \sim \mathcal{D}} \Big[\big(f_\theta(\mathbf{x}, \mathbf{e}, \mathbf{y}^{\text{gen}}) - s \big)^2 \Big], \tag{10}$$

where $\mathcal{D}$ is the dataset constructed from both $\mathcal{D}_{\text{ps}}$ and $\mathcal{D}_{\text{pt}}$. We use multiple checkpoints of the trained retrieval module to obtain various $\mathbf{y}^{\text{gen}}$ to better train the regression model.

Expermental Details. We use the open-source instruction fine-tuned LLMs as backbone. That is Qwen-2.5-14b[6] and LLama-3.3-70b-4bit[7]. We also use several LLMs with commercial API, that is ChatGPT-3.5[8], GPT4o[9] and DeepSeek[10]. The temperature is set to be 0.4. We set the hyperparameters with the KPTimes validation dataset. The learning rate is 3e-5 and the optimizer is AdamW. We retain only the top 2 demonstrations to prompt the LLMs, avoiding that an excessive number of examples would introduce significant additional overhead. We train the model on 2 A40 48GB GPUs and use a cosine scheduler with a 2% warm-up period for 3 epochs. The ratio α in Eq.9 is set to be 0.8, the global threshold γ_1 is set to be 0.70.

A.2 Ablation Study

We perform comprehensive ablation studies based on the DeepSeek model to show the effectiveness of the components through the performance on the three different domain test datasets.

- Zero-shot: This variant only generates keyphrases without demonstrations.

[1] https://github.com/facebookresearch/DPR/

[2] We use the NLTK toolkit to do processing. https://www.nltk.org/

[3] https://huggingface.co/thenlper/gte-large

[4] https://scikit-learn.org/

[5] https://huggingface.co/FacebookAI/roberta-base

[6] https://huggingface.co/Qwen/Qwen2.5-14B-Instruct

[7] https://huggingface.co/ibnzterrell/Meta-Llama-3.3-70B-Instruct-AWQ-INT4

[8] gpt-3.5-turbo-0613

[9] gpt-4o-2024-11-20

[10] deepseek/deepseek-chat

- Random-D: This variant randomly selects samples from the KP20k train dataset to prompt the DeepSeek model.

- Embed-D: This variant solely uses semantic similarity to select samples from the KP20k train dataset to prompt the DeepSeek model.

- w/o MMD: this variant removes the MMD loss during the training of the retrieval module.

- w/o Orthogonality: this variant removes the Orthogonality loss during the training of the retrieval module.

- w/o MDD + Orthogonality: This variant only uses contrastive loss to train the retrieval module.

From Table 3, we can conclude that: The performance of zero-shot is the worst, indicating that while large language models exhibit strong generalization ability, they struggle to leverage this advantage when facing samples from new domains. The performance of random-D shows only a limited improvement over zero-shot, suggesting that inaccurate prompts are ineffective in enhancing the performance of large language models. Embed-D performs better than random-D. It indicates that semantically similar samples provide meaningful assistance to large language models.

Compared to the Embed-D method, the results of w/o MMD, w/o Orthogonality, and w/o MMD+Orthogonality are even better, highlighting the effectiveness of the proposed "seek rational demonstration" approach in this paper. This approach, through the retrieval and evaluation modules, is able to provide large language models with the most reasonable sample examples, thereby significantly enhancing their unsupervised cross-domain keyphrase generation ability.

A.3 In-domain Sample Evaluation

Our SRD method is specifically designed for the unsupervised cross-domain KPG task. For small-scale labeled target domain samples (e.g., 10 labeled target domain samples), we can directly use the target domain supervised samples as prompts, integrating them into our SRD method. To evaluate the experimental effects, we designed the following experiments with DeepSeek model:

Present keyphrase generation						
Methods	StackExchange		KPTimes		KPBiomed	
	F1@3	F1@5	F1@3	F1@5	F1@3	F1@5
Zero-shot	5.4	3.7	11.0	8.5	18.4	17.2
Random-D	6.3	3.4	17.5	10.5	8.6	18.1
Embed-D	18.9	8.9	23.2	20.8	28.7	25.5
Ours	**21.2**	**14.9**	**27.7**	**22.7**	**31.3**	**30.0**
Ours(w/o MMD)	19.6	10.2	26.8	21.1	27.9	27.3
Ours(w/o Orthogonality)	19.8	12.8	27.4	21.5	30.1	28.3
Ours(w/o MDD+Orthogonality)	18.2	9.4	25.4	17.6	27.4	26.6
Absent keyphrase generation						
Methods	StackExchange		KPTimes		KPBiomed	
	R@5	R@10	R@5	R@10	R@5	R@10
Zero-shot	3.1	3.2	5.8	6.0	2.9	2.9
Random-D	3.7	3.2	5.7	5.9	3.2	2.9
Embed-D	6.6	7.1	7.4	9.2	3.8	3.1
Ours	**8.7**	**9.4**	**10.9**	**11.1**	**4.6**	4.8
Ours(w/o MMD)	7.5	7.7	9.2	9.4	3.9	4.6
Ours(w/o Orthogonality)	8.1	7.5	9.8	10.5	4.2	**5.1**
Ours(w/o MDD+Orthogonality)	6.8	6.5	8.4	8.9	4.2	4.3

Table 3: Ablation study of our SRD approach across three benchmark datasets. Best scores in **bold**, second-best underlined.

Present keyphrase generation							
Methods	StackExchange		KPTimes		KPBiomed		Average
	F1@3	F1@5	F1@3	F1@5	F1@3	F1@5	
Ours	21.2	14.9	27.7	22.7	31.3	30.0	24.63
Random-T	22.1	15.3	27.5	23.6	30.8	29.9	24.87
Ours+few-shot	**23.2**	**16.5**	**29.5**	**24.4**	**33.1**	**31.6**	**26.38**
Absent keyphrase generation							
Methods	StackExchange		KPTimes		KPBiomed		Average
	R@5	R@10	R@5	R@10	R@5	R@10	
Ours	8.7	9.4	10.9	11.1	4.6	4.8	8.25
Random-T	8.6	10.1	10.5	11.8	4.5	5.7	8.53
Ours+few-shot	**10.5**	**11.2**	**11.7**	**12.5**	**6.3**	**6.5**	**9.78**

Table 4: In-domain evaluation across three benchmark datasets.

- Random-T: It selects the first 10 samples from the validation sets of their respective datasets, then randomly choose 2 samples as target domain demonstrations to evaluate the generative capability of the LLMs.

- Ours+few-shot: We add 2 randomly selected labeled target domain samples as demonstrations.

From Table 4, we can observe that: The SRD is slightly inferior to supervised target domain KPG, indicates that demonstrations from the same domain have a significant impact on the KPG via LLMs. The SRD+few-shot yields the best results, suggesting that equipping SRD with labeled target domain samples further enhances its KPG capabilities. We infer that, in the context of unsupervised cross-domain KPG, the absence of relevant target domain information usually make it challenging to surpass supervised cross-domain KPG tasks. The assistance of a few-shot approach undoubtedly enhances our method significantly.

A.4 MMD for evaluating distribution shift between two different datasets

The maximum mean discrepancy (MMD) is a measure of the difference between distributions $\mathcal{P}$ and

$\mathcal{Q}$, given by the supremum over a function space $\mathcal{F}$ of differences between the expectations with respect to two distributions. Given n samples from $\mathcal{P}$ as $\mathrm{X} = \{x_i \sim P, i \in [n]\}$, and m samples from $\mathcal{Q}$ as $\mathbf{Z} = \{z_i \sim Q, i \in [m]\}$, the following is a finite sample approximation:

$$\begin{aligned}\mathrm{MMD}(\mathcal{F}, P, Q) = \sup_{f \in \mathcal{F}} (&\mathrm{E}_{X \sim P}[f(X)] \\ &- \mathrm{E}_{Y \sim Q}[f(Y)]).\end{aligned} \tag{11}$$

Inspired by the sampling technique for interpretable machine learning (Kim et al., 2016), we adopt the squared maximum mean discrepancy (MMD) between $\mathcal{S}$ and $\mathcal{P}$ with a kernel function k to measure the discrepancy between them:

$$\begin{aligned}\mathrm{MMD}_k^2(\mathcal{F}, \mathcal{S}, \mathcal{P}) = &\frac{1}{|\mathcal{S}|^2} \sum_{\mathbf{s}_i, \mathbf{s}_j \in \mathcal{S}} k(\mathbf{s}_i, \mathbf{s}_j) \\ &- \frac{2}{|\mathcal{S}||\mathcal{P}|} \sum_{\mathbf{s}_i \in \mathcal{S}, \mathbf{p}_j \in \mathcal{T}} k(\mathbf{s}_i, \mathbf{p}_j) \\ &+ \frac{1}{|\mathcal{P}|^2} \sum_{\mathbf{p}_i, \mathbf{p}_j \in \mathcal{P}} k(\mathbf{p}_i, \mathbf{p}_j).\end{aligned} \tag{12}$$

It is clear that $\mathrm{MMD}^2(\mathcal{F}, P, Q) \geq 0$ and $\mathrm{MMD}^2(\mathcal{F}, P, Q) = 0$ iff. $\mathcal{P}$ is indistinguishable from $\mathcal{Q}$ on the RHKS $\mathcal{F}$.

When $\mathcal{F}$ is a reproducing kernel Hilbert space (RKHS) with kernel function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, the supremum is achieved at (Gretton et al.):

$$f(x) = \mathrm{E}_{X' \sim P}\left[k\left(x, X'\right)\right] - \mathrm{E}_{X' \sim Q}\left[k\left(x, X'\right)\right], \tag{13}$$

and the witness function is approximated as:

$$f(x) = \frac{1}{n} \sum_{i \in [n]} k(x, x_i) - \frac{1}{m} \sum_{j \in [m]} k(x, z_j). \tag{14}$$

The function 14 is also known as the witness function as it measures the maximum discrepancy between the two expectations in $\mathcal{F}$.

A.5 Prompt Templates

For the zero-shot setting of LLMs for unsupervised cross-domain PKG, we use the following prompt template.

```
=======
Please generate accurate present and
absent keyphrases in lowercase for
the given sample.
=======
###Sample###:
"text": "{text}"
**Please only output at least five
present and absent keyphrases with
standard JSON format.**
```

Given the retrieved demonstrations, we can use the following prompt template for unsupervised cross-domain PKG.

```
=======
Please refer to the demonstrations
generate accurate present and absent
keyphrases with lowercase for the given
sample.
{demonstrations_case}
=======
###Sample###:
"text": "{text}"
The MMD distances between the
demonstrations and the sample are {MMD}.
**Only output present and absent
keyphrases with standard JSON format.
```

LexKeyPlan: Planning with Keyphrases and Retrieval Augmentation for Legal Text Generation: A Case Study on European Court of Human Rights Cases

Santosh T.Y.S.S, Elvin Quero Hernandez
School of Computation, Information, and Technology
Technical University of Munich, Germany
{santosh.tokala, elvin.quero}@tum.de

Abstract

Large language models excel at legal text generation but often produce hallucinations due to their sole reliance on parametric knowledge. Retrieval-augmented models mitigate this by providing relevant external documents to the model but struggle when retrieval is based only on past context, which may not align with the model's intended future content. We introduce LexKeyPlan, a novel framework that integrates anticipatory planning into generation. Instead of relying solely on context for retrieval, LexKeyPlan generates keyphrases outlining future content serving as forward-looking plan, guiding retrieval for more accurate text generation. This work incorporates planning into legal text generation, demonstrating how keyphrases—representing legal concepts—enhance factual accuracy. By structuring retrieval around legal concepts, LexKeyPlan better aligns with legal reasoning, making it particularly suited for legal applications. Using the ECHR corpus as case study, we show that LexKeyPlan improves factual accuracy and coherence by retrieving information aligned with the intended content.

1 Introduction

Recent advancements in large language models (LLMs) have led to their adoption in the legal domain for tasks such as sifting through case briefs, expediting legal research, drafting contracts, and formulating litigation strategies (Dahl et al., 2024). Despite their ability to pass bar and law school exams (Katz et al., 2024; Martínez, 2024), perform statutory reasoning and interpretation (Blair-Stanek et al., 2023; Engel and Mcadams, 2024), and apply legal reasoning frameworks like Issue-Rule-Application-Conclusion (IRAC) (Kang et al., 2023; Guha et al., 2024), LLMs still struggle with factual inaccuracies and hallucinations of legal knowledge.

Traditional LLMs generate responses based on the input context and their parametric knowledge (Radford et al., 2019; Zhang et al., 2022; Jiang et al., 2023; Touvron et al., 2023), but this approach often fails to ensure factual accuracy. Retrieval-augmented generation (RAG) addresses this issue by retrieving external documents based on the context, allowing the model to condition its responses on more reliable information (Lewis et al., 2020; Izacard et al., 2023; Borgeaud et al., 2022; Ram et al., 2023). However, RAG relies entirely on the retrieval mechanism's ability to fetch relevant knowledge, which may not always align with the content the model intends to generate. This limitation is particularly critical in legal text generation, where context alone may not provide sufficient signals to retrieve legally relevant information (Magesh et al., 2024; Santosh et al., 2025).

To overcome this challenge, we propose LexKeyPlan, a novel framework that introduces an anticipatory planning stage into the generation process. Instead of relying solely on context for retrieval, LexKeyPlan first generates keyphrases that represent legal concepts outlining the intended content of the response. These keyphrases serve as a forward-looking content plan, guiding the retrieval of relevant documents from external sources. By shifting the retrieval process from purely reactive to proactively structured around anticipated content, LexKeyPlan better aligns retrieval with legal reasoning, reduces reliance on misleading contextual information, and improves coherence in long-form legal text generation.

This work incorporates content planning into legal text generation, highlighting how keyphrases—representing legal concepts—can enhance retrieval, factual grounding, and structured reasoning. Using the European Court of Human Rights (ECHR) case law corpus, we demonstrate that LexKeyPlan significantly improves legal text generation by reducing hallucinations and enhancing retrieval relevance. Our findings suggest that anticipatory content planning through legal con-

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 425–436
July 27 - August 1, 2025

cepts or keyphrases is crucial for legal applications, where precise retrieval and structured reasoning are essential for generating legally sound text.

2 LexKeyPlan

We propose a three-step framework for legal text generation, LexKeyPlan. Unlike traditional models that generate responses directly from context, LexKeyPlan introduces an intermediate planning stage using keyphrases. These keyphrases serve as a forward-looking blueprint, outlining the intent of the future response. By anticipating the necessary content, they guide the retrieval of relevant external documents, ensuring that the generated text is both factually accurate and grounded. This approach contrasts with existing retrieval augmentation methods, which typically use context alone to retrieve relevant information, potentially limiting the relevance of the retrieved content for response generation, in long form text generation.

Formally, let x be the input context. The model first generates a content plan c for the response, which can be represented as $p(c \mid x)$. This content plan, consisting of keyphrases, succinctly captures the essence of the future response. Then the generated plan c is used as query to retrieve relevant documents d from external knowledge sources. The final response y is produced by conditioning the model on the input context, the content plan and the retrieved documents, represented as $p(y \mid c, p, d)$.

Training the Content Plan Generation Module: We train a language model using the input context followed by the content plan. To obtain the target content plan for each input, we utilize the target response text and extract keyphrases from it. We explore two keyphrase extraction algorithms to create the supervision dataset for training this content plan generation module: TextRank (Mihalcea and Tarau, 2004), a graph-based ranking algorithm inspired by PageRank, and KeyBERT (Grootendorst, 2020), leveraging embeddings from pre-trained models.

Retriever Module: We investigate two off-the-shelf retrievers: 1) BM25 (Robertson et al., 2004), a sparse lexical-based model, and 2) GTR (Ni et al., 2021), an embedding-based dense retriever model built on T5-XXL (Raffel et al., 2020). We provide the concatenated list of keyphrases (content plan) as query to retrieve the relevant documents from datastore, built from ECHR judgement documents.

Training the Response Generation Module: The language model is trained using the input context, content plan, retrieved relevant documents followed by the target response. Since we do not have a gold-standard content plan or relevant documents for training, we extract keyphrases from the target response to serve as the content plan and use the target response as a query to retrieve pseudo-golden relevant documents. During inference, the model uses the actual generated content plan to retrieve relevant documents.

This three-step framework offers controllability, enabling users to adjust the content plan or retrieved documents to steer the model's response.

3 Experiments

3.1 Dataset & Metrics

ECHR CaseLaw consists of case judgments heard by the European Court of Human Rights. We use the latest cleaned version of this dataset from Santosh et al. (2024a), which contains 15,729 cases in English from 1960 to July 28, 2022. We partition the dataset chronologically into training, validation, and test splits, following the provided temporal boundaries: the training set includes 11,615 documents from 1960 to 2016, the validation set contains 1,688 documents from 2017–2018, and the test set comprises 2,426 judgments spanning 2018–2022. Each case is segmented into two main sections: *The Facts*, which details the factual background of each case and *The Law*, which presents the legal reasoning justifying the outcome concerning alleged violations of specific ECHR articles. Our task involves generating the *The Law* (reasoning) section based on the *The Facts* section. While generating an entire section would be ideal, evaluating such large outputs against reference texts poses significant challenges. Therefore, we focus on paragraph-level generation to facilitate more accurate evaluations. Specifically, each paragraph in the *The Law* section serves as the target response, while the input context consists of the *The Facts* section and all preceding paragraphs from *The Law* section up to the target paragraph. This paragraph-level setup results in 82,597 training samples, 12,709 validation samples, and 17,470 test samples. During training, each paragraph is appended with an $< |end\_of\_paragraph| >$ token, allowing us to generate the next paragraph up to this marker during open-ended generation. We use `Mistral-7B-Instruct-v0.1` (Jiang et al., 2023) as our base LM for both plan and response generator and follow parameter efficient fine-tuning using

	Plan Generator	Retriever	R-1 / R-2 / R-L	BERT Score	Align Score	Coh. / Flu.
a	-	-	0.32 / 0.13 / 0.21	0.77	0.52	0.69 / 0.61
b	TextRank	-	0.33 / 0.13 / 0.22	0.72	0.54	0.74 / **0.66**
c	KeyBERT	-	0.34 / 0.14 / 0.21	0.74	0.58	0.72 / 0.64
d	-	BM25	0.37 / 0.14 / 0.23	0.78	0.64	0.63 / 0.61
e	-	GTR	0.37 / 0.15 / 0.24	0.79	0.66	0.64 / 0.63
f	TextRank	BM25	0.37 / **0.17** / **0.26**	0.79	0.7	0.72 / 0.65
g	TextRank	GTR	0.39 / 0.15 / 0.24	0.78	0.72	0.75 / 0.65
h	KeyBERT	BM25	**0.40** / 0.15 / 0.23	0.78	0.73	0.74 / **0.66**
i	KeyBERT	GTR	0.39 / 0.16 / 0.24	**0.80**	**0.75**	**0.76** / 0.65

Table 1: Comparison of LexKeyPlan with Baseline Approaches. The "Plan Generator" column specifies the keyphrase extraction algorithm used to generate supervision signals for training both the planning and response generation modules. The "Retriever" column indicates the retrieval method employed during inference and for obtaining relevant documents during training of the response generator. Note that the keyphrase-based content plan is automatically generated by the model during inference.

LoRA (Hu et al., 2021). Implementation details are provided in App. B.

We evaluate the quality of the generated paragraphs: ROUGE-1,2,L (Lin, 2004) for lexical overlap with the reference paragraph, BERTScore (Zhang et al., 2019) for semantic similarity between generated and reference paragraph, AlignScore (Zha et al., 2023) for factual consistency based on a unified alignment function between the reference and generated text and UniEval (Zhong et al., 2022) that evaluates coherence and fluency of the generated paragraph with respect to the context.

3.2 Results

To demonstrate the effectiveness of LexKeyPlan, we compare our method with several baselines: (a) traditional fine-tuning without keyphrase-based content planning or retrieval augmentation; (b, c) models with keyphrase-based planning, followed by response generation using the context and the generated plan. Two keyphrase extraction algorithms are used as supervision signals for training the planning and response generators; (d, e) models without keyphrase planning that directly retrieve relevant documents based on context and generate responses using the retrieved documents. The response generation models are trained with supervision from their respective retrievers; (f, g, h, i) our three-staged LexKeyPlan method, with different combinations of plan generators and retrievers for training the response generation.

Effect of Keyphrase-based Content Planning: As seen in Table 1, keyphrase-based content planning (b, c) outperforms traditional fine-tuning (a) in terms of coherence, fluency, and faithfulness (AlignScore), while remaining comparable on lexical similarity (ROUGE). This keyphrase planning mechanism allows the model to anticipate and structure its response around key concepts in the upcoming text. By separating the planning and generation stages, the model's cognitive load is reduced, as it no longer needs to generate relevant information and maintain coherence simultaneously. Instead, it can focus on each task independently, resulting in more coherent and contextually accurate outputs. Additionally, this structured planning mitigates hallucinations by providing a clear blueprint for the text, helping the model stay grounded in the key content and reducing the likelihood of producing irrelevant information. Between the two keyphrase algorithms, KeyBERT outperforms on content-related metrics, while TextRank excels in stylistic aspects like coherence and fluency.

Effect of Retrieval Augmentation: Retrieval augmentation (d, e), which utilizes context as the query, effectively addresses hallucination issues by grounding the generated text in external knowledge, resulting in improved ROUGE scores and faithfulness (AlignScore). However, this approach also leads to a decrease in coherence and fluency, likely due to the model's challenge in integrating necessary information from the text while maintaining coherent output. Among the retrieval methods, embedding-based GTR consistently outperforms BM25 across all metrics, revealing the limitations of lexical-based retrievers in capturing semantic nuances. Nonetheless, the smaller gap for BM25 highlights the high lexical overlap and repetitive nature of legal concepts in the documents (Santosh et al., 2024a), making BM25 a scalable and efficient option for legal text retrieval.

LexKeyPlan: Integrating keyphrase-based content planning with retrieval augmentation significantly improves both content quality and stylis-

	Plan	Retreiver	R-1 / R-2 / R-L	BERT Score	Align Score	Coh. / Flu.
a	-	-	0.23 / 0.06 / 0.15	0.72	0.44	0.70 / 0.64
b	-	BM25	0.25 / 0.08 / 0.15	0.74	0.56	0.60 / 0.62
c	-	GTR	0.25 / 0.08 / 0.17	0.75	0.58	0.62 / 0.62
d	✓	-	0.24 / 0.08 / 0.16	0.74	0.50	0.72 / 0.62
e	✓	BM25	0.26 / 0.10 / 0.18	0.76	0.59	0.58 / 0.61
f	✓	GTR	0.27 / 0.10 / 0.16	0.75	0.58	0.59 / 0.60

Table 2: Effect of Integrating Keyphrase-Based Content Planning and Retrieval Augmentation in Zero-Shot Setting. The 'Plan' column specifies whether the model is prompted to generate keyphrase-based content plan for the next response. The Retriever column identifies the retrieval method employed.

	Similarity	Exact Match
Zeroshot	0.67	0.18
Fine-tuned with TextRank	0.72	0.36
Fine-tuned with KeyBERT	0.78	0.42
TextRank (Oracle)	0.96	1.0
KeyBERT (Oracle)	0.95	1.0

Table 3: Evaluation of Content Plan Quality.

tic metrics (statistically significant improvements as per Wilcoxon signed-rank test at a 95% confidence interval over the baselines without planning or retrieval augmentation). By using keyphrases as queries to anticipate forthcoming text, LexKeyPlan offers two key advantages. First, it enhances retrieval accuracy by focusing on key concepts, leading to more relevant documents compared to using context alone, which may miss crucial information for the anticipated text. Second, separating the planning and generation tasks allows the model to produce more coherent text. Rather than performing retrieval, integration, and generation simultaneously, LexKeyPlan enables the model to first plan around keyphrases, thereby reducing cognitive load and resulting in more coherent and contextually accurate outputs. Among the methods evaluated, KeyBERT provided superior supervision for plan generation, while GTR outperformed BM25, highlighting the benefits of embedding-based models over traditional lexical approaches.

Zero-shot Experiments: To evaluate whether integrating content planning and retrieval augmentation into pre-trained models in a zero-shot setting enhances performance, we compare several approaches in Table 2: (a) prompting the pre-trained model to generate a response based solely on the context; (b, c) prompting the model to generate a response based on both the context and relevant documents retrieved from the context; (d) prompting the model to first create a content plan in the form of keyphrases and then generate a response based on this plan; (e, f) prompting the model to generate a content plan, use this plan to retrieve relevant documents, and then generate a response based on both the plan and documents.

As expected, retrieval augmentation (b, c) directly enhances faithfulness (AlignScore) and content quality but results in a decrease in coherence. This drop underscores the difficulty of maintaining coherence while incorporating relevant information. Introducing content planning (d) helps the model manage these tasks by breaking them into sequential steps, which enhances both faithfulness and coherence. When combining content planning with retrieval augmentation (e, f), there is a slight improvement in content quality metrics, though stylistic scores decrease. This suggests that zero-shot models may face challenges with the complexity of handling multiple instructions—such as following a content plan and integrating relevant information.

Analysis of Keyphrases in Content Planning: We evaluate the quality of keyphrases generated through (a) Zero-shot prompting the pre-trained model, and (b, c) using a fine-tuned model supervised with TextRank and KeyBERT, by computing the embedding similarity of each keyphrase to the actual target (i.e., the next paragraph). We report both average similarity and exact match (if the keyphrase appears in the target). Additionally, we present oracle metrics (d, e), where keyphrases are directly extracted from the target paragraph using TextRank and KeyBERT. Results show that fine-tuning with KeyBERT-extracted keyphrases leads to more relevant keyphrase generation with respect to future content, providing informed plans for document retrieval and response generation.

4 Conclusion

We introduce LexKeyPlan, a three-step framework for legal text generation integrating keyphrase-based content planning with retrieval augmentation. Unlike traditional methods that rely solely on context for retrieval and generation, LexKeyPlan first

generates a content plan outlining the intended future content. This plan not only guides to produce coherent response but also the retrieval of relevant documents. LexKeyPlan through its anticipatory planning ensures that the generated text is both contextually coherent and factually accurate. as demonstrated with the ECHR case law corpus.

Limitations

LexKeyPlan currently utilizes a relatively simple content planning approach based on keyphrase extraction using general-purpose algorithms such as TextRank and KeyBERT. These methods, while effective in domains like news and scientific literature, were not specifically developed for legal language. As a result, they may fail to capture the nuanced terminologies, hierarchical concepts, and contextual dependencies intrinsic to legal texts. Future work could explore legal-domain-specific keyphrase extraction techniques—such as those proposed in Mandal et al. (2017)—which better accommodate the structural and semantic idiosyncrasies of legal discourse.

Our use of general extractors and retrievers was intentional: we aimed to demonstrate LexKeyPlan's core contribution—anticipatory retrieval guided by future intent—without conflating gains from domain adaptation. Employing legal-specific strategies such as regular-expression-based legal phrase extractors or retrievers fine-tuned on legal corpora may have artificially inflated performance, obscuring the effectiveness of our planning and retrieval coordination. Instead, we used BM25 and GTR as retreivers in addition to TextRank and KeyBERT as keyphrase extractors to validate that our approach can generalize across jurisdictions and is not overly reliant on domain-specific tuning. That said, this choice also introduces limitations. General retrievers often struggle with legal-specific vocabulary, temporal relevance, and the authoritativeness of retrieved materials. In legal contexts, it is not enough to retrieve documents that are merely topically related—their precedential weight, binding status, and jurisprudential relevance must also be considered. For instance, a retrieved case may be outdated, overruled, or of low precedential value, which undermines its utility in supporting legal reasoning. These subtleties are particularly critical in legal domains, where the quality and authority of supporting material can significantly affect argument strength. Future work should explore authority-aware and time-sensitive retrieval mechanisms, potentially through metadata-aware filtering, precedent strength modeling, or temporal reranking that prioritizes current and legally influential sources. (Santosh et al., 2024c,b).

More sophisticated content planning approaches could significantly enhance the effectiveness of LexKeyPlan. One promising avenue is the use of graphical representations derived from legal concept networks. These networks visually map out legal concepts and their interrelationships, creating a structured framework that can guide content generation. Incorporating a specialized legal thesaurus into the planning process can further refine content generation. A legal thesaurus provides a curated vocabulary of legal terms and concepts, enhancing the precision of keyphrases and improving the contextual relevance of the generated content. Improving the integration of content plans during the text generation phase may also involve adopting constrained decoding techniques. Constrained decoding methods enforce adherence to predefined content plans, ensuring that the generated text aligns closely with the planned structure and content. This approach can help maintain coherence and fidelity, reducing deviations from the intended content and improving output quality.

Currently, LexKeyPlan's performance is evaluated using established metrics like ROUGE, BERTScore, and AlignScore. While these metrics offer quantitative insights, they may not fully capture the complexities of legal content. Future research could develop domain-specific evaluation metrics tailored to the legal field. Another limitation of our study is the absence of direct validation by legal experts in the assessment of outputs, which we could not perform due to lack of access to legal experts. Lastly, we selected the ECHR dataset for evaluation due to its explicit segmentation between case facts and legal reasoning. This clean structure enabled a well-defined simulation of a real-world task: generating legal reasoning given only the case facts. Such a setup mirrors copilot-style applications where LLMs assist lawyers and judges in drafting legal arguments. Other legal datasets often lack this granularity, intermixing facts and reasoning, which makes it difficult to isolate and measure the model's reasoning capabilities. However, this focus on ECHR also narrows the scope of our evaluation. Future research should assess LexKeyPlan's adaptability across multiple legal systems and languages, including those with different document

structures and procedural norms.

Ethics Statement

The ECHR dataset used in this study is publicly available and widely employed in legal NLP research. However, it is important to note that this dataset is not anonymized and contains real names and details of the parties involved in the cases. Although we do not foresee direct harm resulting from our experiments, handling such sensitive data necessitates careful consideration of privacy and ethical implications. Researchers must be diligent in protecting individual privacy and managing sensitive information responsibly, even when using publicly accessible datasets.

LLMs utilized in legal contexts have significant implications for both legal professionals and the public. Despite their advanced capabilities, these models carry inherent risks. They may reproduce and amplify biases inherent in their training data, leading to outputs that reflect historical inequalities or stereotypes. Additionally, LLMs may produce incorrect or misleading information, particularly in complex legal scenarios where precision is essential. Our research focuses on enhancing rather than replacing human expertise. It is crucial to apply LLM-generated outputs with careful scrutiny, ensuring that these tools are complemented by thorough human oversight and critical evaluation. Our aim is to improve legal text generation while recognizing that AI must be a supportive tool, used with an awareness of its limitations and potential impact on legal practice.

References

Griffin Adams, Alexander R Fabbri, Faisal Ladhak, Kathleen McKeown, and Noémie Elhadad. 2023. Generating edu extracts for plan-guided summary re-ranking. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2023, page 2680. NIH Public Access.

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization.

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. 2023. Can gpt-3 perform statutory reasoning? In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 22–31.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. Large legal fictions: Profiling legal hallucinations in large language models journal of legal analysis (forthcoming).

Christoph Engel and Richard H Mcadams. 2024. Asking gpt for the ordinary meaning of statutory terms. *MPI Collective Goods Discussion Paper*, (2024/5).

Ameya Godbole, Nicholas Monath, Seungyeon Kim, Ankit Singh Rawat, Andrew Mccallum, and Manzil Zaheer. 2024. Analysis of plan-based retrieval for grounded text generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13101–13119.

Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, et al. 2024. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in Neural Information Processing Systems*, 36.

Markus Guhe. 2020. *Incremental conceptualization for language production*. Psychology Press.

Eduard H Hovy. 1993. Automated discourse generation using discourse structure relations. *Artificial intelligence*, 63(1-2):341–385.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Fantine Huot, Joshua Maynez, Shashi Narayan, Reinald Kim Amplayo, Kuzman Ganchev, Annie Priyadarshini Louis, Anders Sandholm, Dipanjan Das, and Mirella Lapata. 2023. Text-blueprint: An interactive platform for plan-based conditional generation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 105–116.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Xiaoxi Kang, Lizhen Qu, Lay-Ki Soon, Adnan Trakic, Terry Yue Zhuo, Patrick Charles Emerton, and Genevieve Grant. 2023. Can chatgpt perform reasoning using the irac method in analyzing legal scenarios like a lawyer? *arXiv preprint arXiv:2310.14880*.

Nikiforos Karamanis. 2004. Entity coherence for descriptive text structuring.

Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. 2024. Gpt-4 passes the bar exam. *Philosophical Transactions of the Royal Society A*, 382(2270):20230254.

Rodger Kibble and Richard Power. 2004. Optimizing referential coherence in text generation. *Computational Linguistics*, 30(4):401–416.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Anirban Laha, Parag Jain, Abhijit Mishra, and Karthik Sankaranarayanan. 2020. Scalable micro-planned generation of discourse from structured data. *Computational Linguistics*, 45(4):737–763.

Willem JM Levelt. 1993. *Speaking: From intention to articulation*. MIT press.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Zhengyuan Liu and Nancy F Chen. 2021. Controllable neural dialogue summarization with personal named entity planning. *arXiv preprint arXiv:2109.13070*.

Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D Manning, and Daniel E Ho. 2024. Hallucination-free? assessing the reliability of leading ai legal research tools. *arXiv preprint arXiv:2405.20362*.

Arpan Mandal, Kripabandhu Ghosh, Arindam Pal, and Saptarshi Ghosh. 2017. Automatic catchphrase identification from legal court case documents. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2187–2190.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Eric Martínez. 2024. Re-evaluating gpt-4's bar exam performance. *Artificial Intelligence and Law*, pages 1–24.

Kathleen McKeown and Dragomir R Radev. 1995. Generating summaries of multiple news articles. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 74–82.

Kathleen R McKeown. 1985. Discourse strategies for generating natural-language text. *Artificial intelligence*, 27(1):1–41.

Chris Mellish, Alistair Knott, Jon Oberlander, and Mick O'Donnell. 1998. Experiments using stochastic search for text planning. In *Proceedings of the 9th International General Workshop*, pages 98–107. ACL Anthology.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. *arXiv preprint arXiv:1904.03396*.

Shashi Narayan, Joshua Maynez, Jakub Adamek, Daniele Pighin, Blaz Bratanic, and Ryan McDonald. 2020. Stepwise extractive summarization and planning with structured transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4143–4159.

Shashi Narayan, Joshua Maynez, Reinald Kim Amplayo, Kuzman Ganchev, Annie Louis, Fantine Huot, Anders Sandholm, Dipanjan Das, and Mirella Lapata. 2023. Conditional generation with a question-answering blueprint. *Transactions of the Association for Computational Linguistics*, 11:974–996.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with learned entity prompts for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with entity modeling. *arXiv preprint arXiv:1906.03221*.

Ratish Puduppully, Yao Fu, and Mirella Lapata. 2022. Data-to-text generation with variational sequential planning. *Transactions of the Association for Computational Linguistics*, 10:697–715.

Ratish Puduppully and Mirella Lapata. 2021. Data-to-text generation with macro planning. *Transactions of the Association for Computational Linguistics*, 9:510–527.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Stephen Robertson, Hugo Zaragoza, and Michael Taylor. 2004. Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49.

TYS Santosh, Rashid Gustav Haddad, and Matthias Grabmair. 2024a. Ecthr-pcr: A dataset for precedent understanding and prior case retrieval in the european court of human rights. *arXiv preprint arXiv:2404.00596*.

TYSS Santosh, Kevin D Ashley, Katie Atkinson, and Matthias Grabmair. 2024b. Towards supporting legal argumentation with nlp: Is more data really all you need? In *Proceedings of the Natural Legal Language Processing Workshop 2024*, pages 404–421.

TYSS Santosh, Isaac Misael OlguÃn Nolasco, and Matthias Grabmair. 2025. Lecopcr: Legal concept-guided prior case retrieval for european court of human rights cases. *arXiv preprint arXiv:2501.14114*.

TYSS Santosh, Tuan-Quang Vuong, and Matthias Grabmair. 2024c. Chronoslex: Time-aware incremental training for temporal generalization of legal classification tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3022–3039.

Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. 2024. Assisting in writing wikipedia-like articles from scratch with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6252–6278.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function. *arXiv preprint arXiv:2305.16739*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. *arXiv preprint arXiv:2210.07197*.

Appendix

A Related Work

Content Planning for Text Generation: Structured planning is considered a critical link in organizing content effectively before realization (Reiter and Dale, 1997), much like humans plan at a higher level than individual words, as evidenced by psycholinguistic studies (Levelt, 1993; Guhe, 2020). Earlier approaches incorporated various planning representations, such as Rhetorical Structure Theory (Mann and Thompson, 1988; Hovy, 1993) and MUC-style representations (McKeown and Radev, 1995), discourse trees (Mellish et al., 1998), entity transitions (Kibble and Power, 2004; Barzilay and Lapata, 2008), sequences of propositions (Karamanis, 2004), schemas (McKeown, 1985) and lexical chains (Barzilay and Elhadad, 1997).

Recent works in the data-to-text generation task divide it into two distinct phases: planning and realization of natural language text (Moryossef et al., 2019). Puduppully et al. (2019) and Laha et al. (2020) proposed micro-planning strategies, where they first establish a content plan based on a sequence of records and then generate a summary conditioned on that plan. Similar planning approaches have been explored for entity realization (Puduppully et al., 2019). Additionally, Puduppully and Lapata (2021) advocated for macro-planning as

a means of organizing high-level document content, either in text form or within latent space (Puduppully et al., 2022).

In summarization, various plan representations have been investigated. Narayan et al. (2020) treat step-by-step content selection as a planning component, generating sentence-level plans through extract-then-abstract methods, and even sub-sentence-level plans using elementary discourse units (Adams et al., 2023). Further research by Narayan et al. (2021); Liu and Chen (2021) introduced intermediate plans using entity chains—ordered sequences of entities mentioned in the summary. More recently, Narayan et al. (2023); Huot et al. (2023) conceptualized text plans as a sequence of question-answer pairs, which serve as blueprints for content selection (i.e., what to say) and planning (i.e., in what order) in summarization tasks.

Recently, Godbole et al. (2024); Shao et al. (2024) have explored planning-based approaches for text generation in tasks such as expository writing and biographical summaries. However, legal text generation presents unique challenges that remain unexplored in this context. Unlike traditional QA tasks, which involve focused information-seeking queries, legal text generation requires reasoning over long, complex case facts to construct well-grounded arguments to derive conclusions. This makes planning particularly valuable, as it can help structure retrieval and generation to ensure coherence, factual consistency, and legal validity. To the best of our knowledge, no prior work has investigated planning-based approaches for legal text generation.

B Implementation Details

We implemented KeyBERT [1] with n-gram range of (1, 3). We enable Maximal Marginal Relevance (MMR) with a diversity value 0.7 to balance relevance and diversity. For TextRank (Mihalcea and Tarau, 2004), we utilize the implementation provided in the GitHub repository.

We perform LoRA fine-tuning with Mistral model for both our plan and response generator with an alpha and rank configuration in $\{8, 16\}$ and dropout of 0.1. We sweep over learning rates $\{1e-5, 3e-5, 5e-5, 1e-4, 3e-4\}$ and evaluating its performance on the validation set using 10K validation steps. We train the model end-to-end for one epoch using Adam Optimizer (Kingma and Ba, 2014). We perform inference over models with a temperature of 0.7 and nucleus sampling of 0.95.

For retrieval, we dynamically populate the candidate datastore by excluding documents dated after the query case and including only those available up to the query date, thereby simulating a realistic setting. We segment these documents into paragraph chunks, to index them for retrieval. For retrieval, we use the top-3 retrieved paragraphs for both methods. Prompt 1 and 2 detail the prompts used for content plan generation and response generations modules respectively.

C Case Study

We present three case studies that demonstrate how LexKeyPlan performs in complex legal scenarios. These examples show how keyphrase planning enables doctrinally grounded retrieval and controllable generation, while also revealing important limitation of its inability to account for temporal relevance.

Case Study A: Enhancing Doctrinal Accuracy Through Keyphrase-Guided Retrieval We examine a scenario where an individual is convicted for inflammatory statements made on social media and brings a claim under Article 10 (freedom of expression). The government argues that the speech in question incited hatred and thus falls outside the protection of the Convention. This case hinges on the distinction between political criticism and hate speech—a critical threshold question that determines whether the speech is covered by Article 10 or excluded under Article 17, which bars the abuse of rights. When the model operates with context-only retrieval, it relies solely on the case facts—namely, that the applicant was prosecuted for a speech act and invokes Article 10. This prompts the retriever to surface general freedom of expression jurisprudence such as Handyside v. UK (1976), Sunday Times v. UK (1979), and Oberschlick v. Austria (1997), all of which affirm the importance of free expression but apply a proportionality framework under Article 10(2). As a result, the generated response reasons as follows: "The Court applies the proportionality test under Article 10(2) to determine whether the restriction was necessary in a democratic society. Given the precedent set in Handyside v. UK, the speech must be examined in light of public morals and state discretion. Therefore, the speech in this case may

[1] https://github.com/MaartenGr/KeyBERT

be protected under Article 10." This response is legally incorrect: it prematurely applies a balancing test without addressing the more fundamental question of admissibility under Article 10. Specifically, if the speech constitutes hate speech or denial of historical atrocities, it may be excluded altogether under Article 17, rendering Article 10 inapplicable. The model's failure to retrieve the right precedents results in doctrinal error. In contrast, with keyphrase-guided planning, the model is given key concepts such as "legitimate aim (public safety vs political criticism)," "ECHR Article 17," "abuse of rights doctrine," and "margin of appreciation." These guide retrieval toward cases that directly address the legal test relevant to hate speech and the abuse of rights. The retrieved case set now includes Garaudy v. France (2003), Perinçek v. Switzerland (2015), and Leroy v. France (2008)—cases that explicitly delineate when speech is excluded under Article 17 or subjected to Article 10 balancing. With this doctrinally appropriate retrieval, the model produces the following improved response: "The Court first determines whether the applicant's speech falls within the scope of Article 10 or is excluded under Article 17, which bars the abuse of Convention rights. In Garaudy v. France, the Court held that speech denying historical atrocities such as the Holocaust is incompatible with the values of the Convention and is not protected under Article 10. If the applicant's statements incite hatred or deny established facts, they may fall under Article 17 and be excluded from protection. However, if the speech qualifies as political criticism—even if provocative—it may still enjoy Article 10 protection, subject to a proportionality assessment under Article 10(2), as seen in Leroy v. France." This output demonstrates significantly improved legal reasoning: it reflects the correct structure of analysis (i.e., first evaluating exclusion under Article 17 before engaging in Article 10 balancing), references the right cases, and avoids premature or inappropriate application of proportionality. It also recognizes the margin of appreciation doctrine and the need to distinguish incitement from legitimate political critique. This case study underscores a core strength of LexKeyPlan: by prompting retrieval based on key doctrinal concepts rather than surface-level case similarity, the model can reason more accurately and in line with legal standards.

Case Study B: Controlling Legal Reasoning via Structured Planning In a case where a political figure challenged the deplatforming of their account by a private social media company, alleging a violation of Article 10. The case raised ambiguity about whether private actors performing public functions are subject to free speech obligations. LexKeyPlan was able to demonstrate controllable reasoning by varying the content plan. When the keyphrases included "de facto public forum," "chilling effect," and "corporate control over speech," the model retrieved cases such as Appleby and Others v. UK (2003), which address the public forum doctrine in private spaces. The generated response argued that: "Social media platforms serve as a de facto public forum, where political discourse primarily unfolds. As established in Appleby v. UK, while private entities are not always bound by public free speech obligations, their restrictions may be scrutinized if they effectively eliminate a person's ability to communicate elsewhere. Given the dominance of social platforms in modern public debate, banning a political figure from these spaces constitutes an undue restriction, creating a chilling effect that stifles democratic discourse." When the content plan was changed to include keyphrases such as "platform autonomy," "harm prevention," "corporate free expression," "content moderation policies," and "disinformation regulation," the retriever instead surfaced Delfi AS v. Estonia (2015), which upholds the rights of private platforms to moderate harmful content. The corresponding generated response shifted the framing: "Social media platforms, as private entities, have the right to moderate content under their terms of service. Prior rulings, including Delfi AS v. Estonia, affirm that private actors are not bound by the same restrictions as state actors. Given the risks of disinformation and harm to democratic integrity, content moderation is a necessary safeguard. The platform's decision aligns with legitimate policy goals and does not constitute a violation of Article 10." This illustrates LexKeyPlan's strength in enabling users to explore competing legal framings by adjusting the planning stage, allowing the system to retrieve and ground responses in support of different legal arguments, rather than being constrained by the framing present in the input context alone.

Case Study C: Limitations in Capturing Temporal Relevance Despite these benefits, LexKeyPlan remains limited in its ability to account for temporal and relevance. In a case involving the extradition of an individual to a country with a poor human rights record, the applicant claimed that the transfer would violate Article 3, citing the

risk of inhuman or degrading treatment. While the keyphrases “inhuman treatment,” “extradition risk,” and “diplomatic assurances” did help retrieve landmark cases like Soering v. UK (1989), they failed to surface more recent jurisprudence such as Othman (Abu Qatada) v. UK (2012) or Paposhvili v. Belgium (2016), which significantly recalibrated the legal standards for assessing risk and evaluating the sufficiency of diplomatic assurances. The limitation here arises from LexKeyPlan’s reliance on topical keyphrase matching without accounting for the temporal dimension of legal knowledge. As legal doctrines evolve, retrieval systems must prioritize not only relevance but recency, particularly in areas where the law is in flux.

```
You are a judge at the European Court of Human Rights (ECHR), drafting a judicial judgment. The
paragraphs below reflect your reasoning thus far. Your task is to outline the next topics you intend
to address in the next paragraph of the judgment.

Instructions:
- Generate a maximum of 5 topics. Place higher-priority topics at the beginning of the list.
- Ensure the list is formatted as a comma-separated string.
- Ensure the topics align closely with the content of the previous paragraphs and the overall context
 of the judgment.

Input:
{{PREVIOUS PARAGRAPHS}}

Response:
```

Prompt 1: Prompt for Keyphrase based content plan generation

```
You are a judge at the European Court of Human Rights (ECHR), drafting a judicial judgment. Below are
 paragraphs you have already written as part of your reasoning. Your task is to write the next
paragraph to continue developing the judgment.

To assist you, content plan for the next pargraph in the form of keyphrases and excerpts from
relevant external cases have been provided. These cases may offer supporting arguments or precedents
to strengthen your reasoning, but their use is optional. Your primary focus should be on ensuring the
 logical flow and coherence of the judgment.

Instructions:

- Refer to external cases only if they add value to your reasoning or support the legal principles
being applied. When citing an external case, include a proper citation at the end. For example: (Case
 of X v. Y). If referencing a specific section or paragraph from an external case, denote it clearly
using the section symbol and the paragraph number.
- Ensure that the new paragraph is directly related to the content of the previous paragraphs and the
 broader context of the ECHR judgment.
- Use formal, precise, and structured language typical of ECHR judgments.


Input:
External Cases:
    {EXTERNAL CASES}

Previous Paragraphs:
    {PREVIOUS PARAGRAPHS}

Response:
```

Prompt 2: Prompt for Response Generation Module

SynWorld: Virtual Scenario Synthesis for Agentic Action Knowledge Refinement

Runnan Fang♠, Xiaobin Wang♡, Yuan Liang♠, Shuofei Qiao♠, Jialong Wu♡, Zekun Xi♠, Ningyu Zhang♠*, Yong Jiang♡*, Pengjun Xie♡, Fei Huang♡, Huajun Chen♠♣*
♠Zhejiang University ♡Alibaba Group
♣Zhejiang Key Laboratory of Big Data Intelligent Computing
{rolnan,zhangningyu}@zju.edu.cn

Abstract

In the interaction between agents and their environments, agents expand their capabilities by planning and executing actions. However, LLM-based agents face substantial challenges when deployed in novel environments or required to navigate unconventional action spaces. To empower agents to autonomously explore environments, optimize workflows, and enhance their understanding of actions, we propose SynWorld, a framework that allows agents to synthesize possible scenarios with multi-step action invocation within the action space and perform Monte Carlo Tree Search (MCTS) exploration to effectively refine their action knowledge in the current environment. Our experiments demonstrate that SynWorld is an effective and general approach to learning action knowledge in new environments[1].

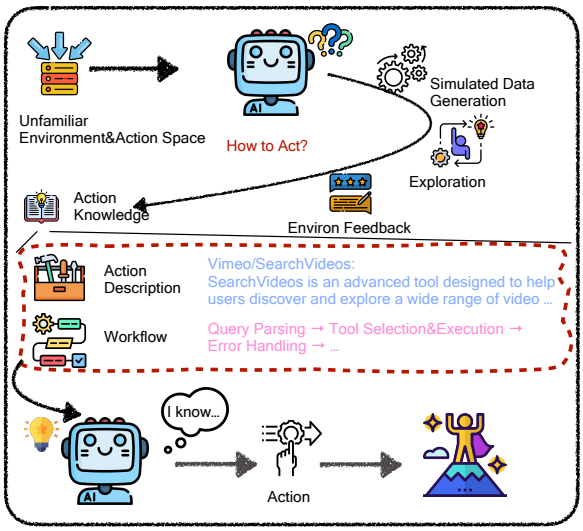


Figure 1: Our method with exploration to refine action knowledge in Synthesized Scenario.

1 Introduction

By leveraging decision-making capabilities to execute task-oriented actions within dynamic environments, Large Language Models (LLM) based agents demonstrate enhanced environmental interactivity and operational versatility (Song et al., 2023a; Liu et al., 2024b; Wu et al., 2025; Xi et al., 2025; Shi et al., 2025; Qu et al., 2025). In the real world, agents perform actions by leveraging tools like web search engines (Fan et al., 2024; Zhao et al., 2024a; Ning et al., 2025) or API calls (Liu et al., 2024a; Tao et al., 2024) to access feedback from the real world, which addresses the static knowledge limitations of LLMs, facilitating a deeper comprehension of the real world. It is crucial for agents to learn how to plan and execute actions in the environment. Nonetheless, as the complexity of tasks increases and novel environments emerge, manually annotated environment descriptions and a predefined action documents (Qu et al., 2024; Sri et al., 2024; Zhang et al., 2025) for agents are often not consistent with the actual environmental conditions and action usage (Liu et al., 2024d; Huang et al., 2024a). Refining well-defined and aligned descriptions of the environment and actions is time-consuming and labor-intensive.

Therefore, to master inexperienced action and complicated task requirements in new complex environments, refinement of the agentic action knowledge is essential. Previous studies have explored the acquisition of action knowledge through feedback in scenarios synthesized by LLMs. Similar to the way humans acquire skills through trial and error, agents can also optimize the descriptions of actions by leveraging feedback from simulated scenarios (Yuan et al., 2024; Du et al., 2024; Bouzenia et al., 2024). However, these methods exhibit two critical limitations: (1) The synthetic scenarios they utilize are often restricted to single-action, which hinders agents from learning workflows suitable for these tasks, and (2) The linear iterative optimization process lacks a clear direction for improvement, making it susceptible to stagnation and

* Corresponding Author.
[1]Code is available at https://github.com/zjunlp/SynWorld.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 437–448
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

quickly reaching its performance ceiling.

To address these limitations, we propose a new framework, **SynWorld**, designed to assist agents in learning unfamiliar actions in new environments as shown in Figure 1. **SynWorld** first synthesizes virtual scenarios involving multiple coordinated actions. Then through iterative MCTS optimization in the exploration of virtual scenarios, the framework enables more thorough and bidirectional refinement between action descriptions and workflow patterns, ensuring better alignment with environmental constraints. Experiments demonstrate that action knowledge can be learned in virtual environments and effectively generalized to the real world, with optimization through MCTS exploration.

2 Background

2.1 Agent Planning

An agent interacts with its environment by perceiving its state, selecting actions to achieve a goal, and learning from feedback in the form of rewards. Its framework consists of a state space $\mathcal{S}$ that represents the environment's properties, an action space $\mathcal{A}$ that defines allowable interactions, and an observation space Ω for perceptual inputs. Progress toward task T is measured through a reward function $\mathcal{R}$. Central to decision-making is a planning mechanism $\mathcal{P}_\theta$, where π_θ are fixed model weights. The agent's architecture is defined by the tuple:

$$\mathcal{P}_\theta = \pi_\theta(\mathcal{S}, \mathcal{A}, \Omega, \mathcal{R}) \tag{1}$$

This formula delineates the manner in which an agent assesses its current state and interprets environmental feedback to generate plans.

2.2 Action Knowledge

Action knowledge $\mathcal{AK}$ serves as the strategic foundation governing an agent's adaptive behavior in dynamic and unfamiliar environments. It contains **action description** about the awareness of executable actions with **cognitive workflows** about task decomposition and action sequences.

3 Method

In this section, we begin by detailing how to utilize the action space to synthesize scenarios and specific objectives. Subsequently, we dive into the application of MCTS to explore and discover action knowledge within these synthesized scenarios. The SynWorld framework is shown in Figure 2.

3.1 Scenario Synthesis

To address generalization challenges in multistep tool operationalization, we propose a framework that synthesizes scenarios through tool-conditioned task generation. Our methodology formalizes scenario synthesis as:

$$\mathcal{S}(t) = \{(\mathcal{B}, \mathcal{G}) \mid \forall t \subseteq T)\}, \tag{2}$$

where a subset of tools t selected by llm from the complete set of tools T to design a scenario. Each scenario comprises two part: **Background** $\mathcal{B}$: The contextual scenario specifying initial conditions and constraints; **Goal** $\mathcal{G}$: The terminal objective requiring tool-mediated resolution. We provide examples using a few-shot approach to enable the llm to synthesize queries.

The mapping enforces that distinct tool combinations yield nontrivial scenario variations through systematic $\mathcal{B}$-$\mathcal{G}$ pairings. Each group of selected tools will generate 2-3 scenarios. To ensure data diversity, if the similarity of a newly generated scenario exceeds a threshold ϵ compared to already synthesized scenarios, it will be excluded. Through this process, we can obtain a large number of synthetic scenarios, where the selected tools will serve as the "gold tools" for completing the corresponding virtual scenario, which will later be used for evaluation purposes.

$$d((\mathcal{B}_i, \mathcal{G}_i), (\mathcal{B}_j, \mathcal{G}_j)) < \epsilon. \tag{3}$$

3.2 Action Knowledge Exploration

Initialization The root node is initialized with predefined Action Knowledge, which serves as the foundation for task-solving logic. During the MCTS process, the UCB algorithm is used to select nodes, effectively balancing exploration and exploitation by choosing the node with the highest upper confidence limit.

Expansion Upon selecting node N_i as the candidate, an optimization process is initiated that retraces N_i to obtain insights from previous optimization experience $\mathcal{E}$. Each of these past optimization experiences $\mathcal{E}$ is composed of three elements: the pre-optimization score S_{before}, the post-optimization score S_{after}, and the modification $\mathcal{M}$ of the optimization actions taken.

$$\mathcal{E} = \{(S^i_{\text{before}}, S^i_{\text{after}}, \mathcal{M}^i) \mid N_i \in \text{Path}(N, N_0)\} \tag{4}$$

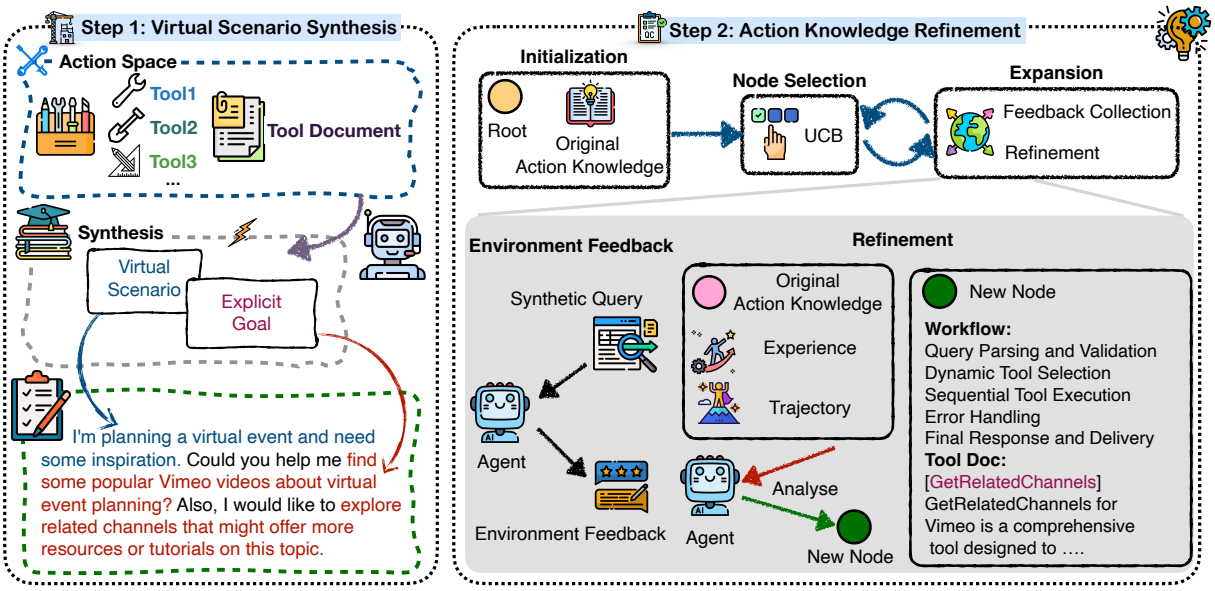


Figure 2: **The overall framework of SynWorld**: we first extract composable tools from the toolkit to generate new scenes and tasks. Then, we allow agents to explore the synthesized virtual scenes using MCTS to optimize action knowledge, thereby learning how to execute actions and plan tasks.

Based on the optimization experiences and exploration trajectories Tra from the past, the LLM-based agent π will analyze the discrepancies between the existing Action Knowledge and the environment. It will then optimize these to produce an updated version of the Action Knowledge.

$$\mathcal{AK}_{new} = \pi_\theta(\mathcal{AK}_{old}, \mathcal{E}, Tra) \quad (5)$$

Feedback Collection Once equipped with an optimized $\mathcal{AK}$, the agent π can explore the environment to perform tasks. For each individual task T, the agent interacts with the environment to receive feedback with the trajectory Tra_i and the final reward scores S_i. The score is related to the evaluation method of the task.

$$Tra_i, S_i = Env(\mathcal{AK}, \pi) \quad (6)$$

4 Experiment

4.1 Experiment Setup

Datasets and Baselines To demonstrate the efficiency of our approach in optimizing action knowledge, we selected two datasets: ToolBench (Qin et al., 2024) and HotpotQA (Yang et al., 2018), each offering unique challenges for a comprehensive evaluation. Following Qu et al. (2024), several strong methods are selected as our baselines, including ReAct (Yao et al., 2023), Self-Refine (Madaan et al., 2023), Easy-Tool (Yuan et al., 2024), and DRAFT (Qu et al., 2024). See detailed setting and evaluation in Appendix B.

Model	Method	ToolBench PASS	ToolBench WIN	HotpotQA
GPT-4-turbo	ReAct	50.67	67.00	54.61
	Self-Refine	56.80	**73.00**	55.85
	EasyTool	51.67	68.00	58.19
	DRAFT	54.83	72.00	57.71
	Ours	**59.33**	**73.00**	**59.93**
Qwen-long	ReAct	48.30	71.00	52.00
	Self-Refine	53.70	77.00	56.10
	EasyTool	50.80	63.00	58.34
	DRAFT	54.20	79.00	53.23
	Ours	**57.20**	**81.00**	**59.91**
Qwen2-72B-Instruct	ReAct	49.43	55.00	50.21
	Self-Refine	54.33	65.00	52.59
	EasyTool	52.97	58.00	54.94
	DRAFT	56.43	69.00	57.57
	Ours	**58.52**	**73.00**	**58.70**

Table 1: **Main results of SynWorld compared to other baselines** on ToolBench and HotpotQA. The best results of each model are marked in **bold**. PASS means the pass rate and WIN means the win rate of the trajectory compared to GPT-3.5-turbo in the method of ReAct.

4.2 Main Results

For the task ToolBench that requires the combined use of multiple tools, as shown in Table 1, our approach achieved a PASS score of 59.33 and a WIN score of 73.00, marking a significant improvement compared to other methods for iterative optimization, demonstrating the advantages of our method in terms of tool combination and task planning optimization. **For the task HotpotQA that requires planning using a single tool and multi-step calls**, in the scenario where only a single tool is used but requires continuous multi-hop calls, our method has achieved state-of-the-art results. This

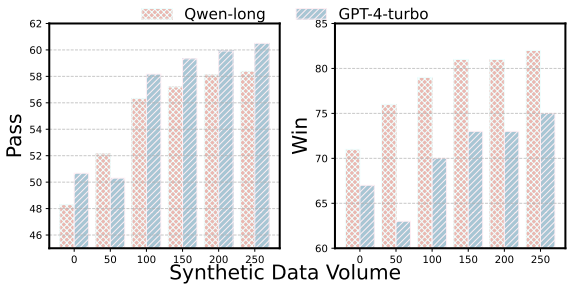


Figure 3: The variation in the pass rate of agents on the ToolBench in relation to the number of exploration scenarios.

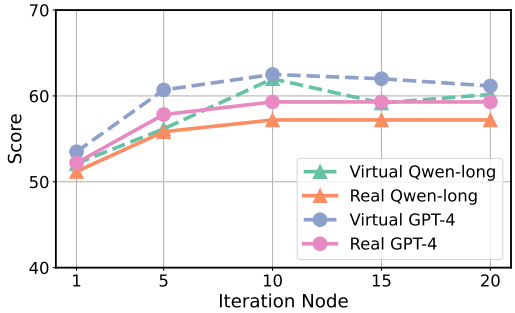


Figure 4: Changes in ToolBench pass rates in virtual and real-world scenarios with the number of iterative optimizations performed in the virtual environment.

indicates that we have not only aligned tool descriptions with the environment, but also succeeded in generating a generalizable planning workflow.

4.3 Ablation Study

We observe that independently optimizing either the Workflow or the Tool Description using MCTS has its limitations in Table 2. We find that combining the optimization of both aspects leads to more effective results. An aligned Tool Description is beneficial for constructing a more reasonable Workflow, while a well-structured, general Workflow also enhances the exploration of tool usage. We believe that this synergy arises during the iterative optimization process, where the improved workflow can help identify tool usage that is closer to the correct trajectory, serving as strong negative examples to further refine the tool description. Conversely, a superior tool description enables the model to generate workflows that are more aligned with the environment.

Model	Method	Pass Rate
GPT-4-turbo	SynWorld	59.33
	w/o. Workflow	56.33-3.00
	w/o. Description	53.16-6.17
Qwen-long	SynWorld	57.20
	w/o. Workflow	57.00-0.20
	w/o. Description	53.83-3.37

Table 2: Ablation experiment results

4.4 Futher Analysis

More simulated data enable precise virtual scenario synthesis, optimized action knowledge, and ultimately improved agent performance. In our experiments, we explore action knowledge by synthesizing a varying number of virtual scenarios. As shown in Figure 3, we find that as the number of scenarios synthesized increases, the performance of the Agent shows a corresponding upward trend. Specifically, within the range of 0 to 100 scenarios, the model's performance continues to improve with the increase in the scenarios, indicating that action knowledge is indeed learnable. Although the rate of performance improvement slows down as the number of scenarios increases, the model's performance remains on an upward trajectory. This phenomenon suggests that the process of learning action knowledge in the context of synthesized scenarios exhibits scalability.

Virtual scenario policies can be generalized to unseen environments and improved with iterations. By analyzing the relationship between action knowledge iterations and pass rates on Toolbench in both virtual and real environments, we find that the action knowledge gained in the virtual setting is generalizable and effective in real-world applications. Performance trends in both environments are similar in Figure 4. We observe a consistent upward trend in scores, particularly between 0 and 10 iterations, indicating that action knowledge can be optimized through environmental feedback. However, as iterations increase, the gains diminish, and we note slight declines in performance at times. This phenomenon is likely due to the limitations of exploring a fixed number of scenarios, where further iterations have less impact, and increasing complexity can hinder understanding.

5 Conclusion

In this paper, we propose SynWorld, a novel framework that synthesizes scenes that require multiple action steps and enhances agent action optimization through exploration in the synthetic virtual scenario. By systematically exploring diverse syn-

thetic scenarios, our model achieves precise alignment between action descriptions and environmental contexts while identifying task-specific workflows suitable for tasks.

Limitations

We initially conduct empirical validation on two benchmarks: Toolbench (involving multi-tool calling scenarios) and HotpotQA (requiring multi-step action execution). While these demonstrate our method's effectiveness, broader validation across diverse real-world applications remains valuable. Promising candidates include web-based search tasks, simulated environments and so on.

Our approach currently incurs non-trivial computational overhead due to the token-intensive virtual scenario synthesis process. The exploration phase further compounds this by exhaustively enumerating all possible scenarios. Future research should prioritize optimizing token efficiency through 1) developing more economical synthesis mechanisms for high-quality virtual scenarios and 2) establishing effective filtering criteria to identify the most pedagogically valuable scenarios.

The current action knowledge representation employs a purely text-based format. This presents opportunities to investigate alternative structured representations that could enhance reasoning capabilities, such as tabular organization of action parameters or executable code snippets encapsulating procedural knowledge.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62206246, No. NSFCU23B2055, No. NSFCU19B2027), the Fundamental Research Funds for the Central Universities (226-2023-00138), Yongjiang Talent Introduction Programme (2021A-156-G), CIPSC-SMP-Zhipu Large Model Cross-Disciplinary Fund, Ningbo Natural Science Foundation (2024J020), Information Technology Center and State Key Lab of CAD&CG, Zhejiang University. We gratefully acknowledge the support of Zhejiang University Education Foundation Qizhen Scholar Foundation.

References

Islem Bouzenia, Premkumar Devanbu, and Michael Pradel. 2024. Repairagent: An autonomous, llm-based agent for program repair. *arXiv preprint arXiv:2403.17134*.

Huajun Chen. 2023. Large knowledge model: Perspectives and challenges. *arXiv preprint arXiv:2312.02706*.

Yu Du, Fangyun Wei, and Hongyang Zhang. 2024. Anytool: Self-reflective, hierarchical agents for large-scale api calls. *arXiv preprint arXiv:2402.04253*.

Zane Durante, Qiuyuan Huang, Naoki Wake, Ran Gong, Jae Sung Park, Bidipta Sarkar, Rohan Taori, Yusuke Noda, Demetri Terzopoulos, Yejin Choi, et al. 2024. Agent ai: Surveying the horizons of multimodal interaction. *arXiv preprint arXiv:2401.03568*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on RAG meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 6491–6501. ACM.

Sihao Hu, Tiansheng Huang, Fatih Ilhan, Selim Tekin, Gaowen Liu, Ramana Kompella, and Ling Liu. 2024. A survey on large language model-based game agents. *arXiv preprint arXiv:2404.02039*.

Jerry Huang, Prasanna Parthasarathi, Mehdi Rezagholizadeh, and Sarath Chandar. 2024a. Towards practical tool usage for continually learning llms. *arXiv preprint arXiv:2404.09339*.

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024b. Understanding the planning of llm agents: A survey. *arXiv preprint arXiv:2402.02716*.

Ziyan Jiang, Xueguang Ma, and Wenhu Chen. 2024. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv preprint arXiv:2406.15319*.

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. 2024. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*.

Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun Zhang, Kaitao Song, Kunlun Zhu, Yuheng Cheng, Suyuchen Wang, Xiaoqiang Wang, Yuyu Luo, Haibo Jin, Peiyan Zhang, Ollie Liu, Jiaqi Chen, Huan Zhang, Zhaoyang Yu, Haochen Shi, Boyan Li, Dekun Wu, Fengwei Teng, Xiaojun Jia, Jiawei Xu, Jinyu Xiang, Yizhang Lin, Tianming Liu, Tongliang Liu, Yu Su, Huan Sun, Glen Berseth, Jianyun Nie, Ian Foster, Logan Ward, Qingyun Wu, Yu Gu, Mingchen Zhuge, Xiangru Tang, Haohan Wang, Jiaxuan You, Chi Wang, Jian Pei, Qiang Yang, Xiaoliang Qi, and Chenglin Wu. 2025. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *Preprint*, arXiv:2504.01990.

Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, Lei Zhang, Jianfeng Gao, and Chunyuan Li. 2024a. Llava-plus: Learning to use tools for creating multimodal agents. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XLVII*, volume 15105 of *Lecture Notes in Computer Science*, pages 126–142. Springer.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024b. Agentbench: Evaluating llms as agents. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yanming Liu, Xinyue Peng, Yuwei Zhang, Jiannan Cao, Xuhong Zhang, Sheng Cheng, Xun Wang, Jianwei Yin, and Tianyu Du. 2024c. Tool-planner: Dynamic solution tree planning for large language model with tool clustering. *arXiv preprint arXiv:2406.03807*.

Zeyu Leo Liu, Shrey Pandit, Xi Ye, Eunsol Choi, and Greg Durrett. 2024d. Codeupdatearena: Benchmarking knowledge editing on API updates. *CoRR*, abs/2407.06249.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tula Masterman, Sandi Besen, Mason Sawtell, and Alex Chao. 2024. The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey. *arXiv preprint arXiv:2404.11584*.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2022. A survey on multi-hop question answering and generation. *arXiv preprint arXiv:2204.09140*.

Liangbo Ning, Ziran Liang, Zhuohang Jiang, Haohao Qu, Yujuan Ding, Wenqi Fan, Xiao-yong Wei, Shanru Lin, Hui Liu, Philip S Yu, et al. 2025. A survey of webagents: Towards next-generation ai agents for web automation with large foundation models. *arXiv preprint arXiv:2503.23350*.

Siqi Ouyang and Lei Li. 2023. Autoplan: Automatic planning of interactive decision-making tasks with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 3114–3128. Association for Computational Linguistics.

Shuofei Qiao, Ningyu Zhang, Runnan Fang, Yujie Luo, Wangchunshu Zhou, Yuchen Eleanor Jiang, Chengfei Lv, and Huajun Chen. 2024. Autoact: Automatic agent learning from scratch via self-planning. *arXiv preprint arXiv:2401.05268*.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. 2024. From exploration to mastery: Enabling llms to master tools via self-driven interactions. *CoRR*, abs/2410.08197.

Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, et al. 2025. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *arXiv preprint arXiv:2503.21614*.

Shreyas Sundara Raman, Vanya Cohen, Eric Rosen, Ifrah Idrees, David Paulius, and Stefanie Tellex. 2022. Planning with large language models via corrective re-prompting. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

Weizhou Shen, Chenliang Li, Hongzhan Chen, Ming Yan, Xiaojun Quan, Hehong Chen, Ji Zhang, and Fei Huang. 2024. Small llms are weak tool learners: A multi-llm agent. *arXiv preprint arXiv:2401.07324*.

Yucheng Shi, Wenhao Yu, Wenlin Yao, Wenhu Chen, and Ninghao Liu. 2025. Towards trustworthy gui agents: A survey. *arXiv preprint arXiv:2503.23434*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. 2021. Alfworld: Aligning text and embodied environments for interactive learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Simranjit Singh, Andreas Karatzas, Michael Fore, Iraklis Anagnostopoulos, and Dimitrios Stamoulis. 2024. An llm-tool compiler for fused parallel function calling. *arXiv preprint arXiv:2405.17438*.

Chan Hee Song, Brian M. Sadler, Jiaman Wu, Wei-Lun Chao, Clayton Washington, and Yu Su. 2023a. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 2986–2997. IEEE.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2023b. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3009.

S Deepika Sri, Raja CSP Raman, Gopinath Rajagopal, S Taranath Chan, et al. 2024. Automating rest api postman test cases using llm. *arXiv preprint arXiv:2404.10678*.

Haotian Sun, Yuchen Zhuang, Lingkai Kong, Bo Dai, and Chao Zhang. 2023. Adaplanner: Adaptive planning from feedback with language models. *Advances in neural information processing systems*, 36:58202–58245.

Chunliang Tao, Xiaojing Fan, and Yahe Yang. 2024. Harnessing llms for api interactions: A framework for classification and synthetic data generation. *arXiv preprint arXiv:2409.11703*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.

Ruoyao Wang, Peter A. Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11279–11298. Association for Computational Linguistics.

Zihao Wang, Shaofei Cai, Anji Liu, Xiaojian Ma, and Yitao Liang. 2023b. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *CoRR*, abs/2302.01560.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025. Webwalker: Benchmarking llms in web traversal. *arXiv preprint arXiv:2501.07572*.

Zekun Xi, Wenbiao Yin, Jizhan Fang, Jialong Wu, Runnan Fang, Ningyu Zhang, Jiang Yong, Pengjun Xie, Fei Huang, and Huajun Chen. 2025. Omnithink: Expanding knowledge boundaries in machine writing through thinking. *Preprint*, arXiv:2501.09751.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Yongliang Shen, Kan Ren, Dongsheng Li, and Deqing Yang. 2024. EASYTOOL: enhancing llm-based agents with concise tool instruction. *CoRR*, abs/2401.06201.

Xilin Zhang, Zhixin Mao, Ziwen Chen, and Shen Gao. 2024. Effective tool augmented multi-agent framework for data analysis. *Data Intelligence*, 6(4):923–945.

Zhuosheng Zhang, Yao Yao, Aston Zhang, Xiangru Tang, Xinbei Ma, Zhiwei He, Yiming Wang, Mark Gerstein, Rui Wang, Gongshen Liu, et al. 2025. Igniting language intelligence: The hitchhiker's guide from chain-of-thought reasoning to language agents. *ACM Computing Surveys*, 57(8):1–39.

Siyun Zhao, Yuqing Yang, Zilong Wang, Zhiyuan He, Luna K Qiu, and Lili Qiu. 2024a. Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely. *arXiv preprint arXiv:2409.14924*.

Suifeng Zhao, Tong Zhou, Zhuoran Jin, Hongbang Yuan, Yubo Chen, Kang Liu, and Sujian Li. 2024b. Awecita: Generating answer with appropriate and well-grained citations using llms. *Data Intelligence*, 6(4):1134–1157.

Yuyue Zhao, Jiancan Wu, Xiang Wang, Wei Tang, Dingxian Wang, and Maarten De Rijke. 2024c. Let me do it for you: Towards llm empowered recommendation via tool learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1796–1806.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Long Li, Jialong Wu, Tiannan Wang, Shi Qiu, Jintian Zhang, Jing Chen, Ruipu Wu, Shuai Wang, et al. 2023. Agents: An open-source framework for autonomous language agents. *arXiv preprint arXiv:2309.07870*.

Wangchunshu Zhou, Yixin Ou, Shengwei Ding, Long Li, Jialong Wu, Tiannan Wang, Jiamin Chen, Shuai Wang, Xiaohua Xu, Ningyu Zhang, et al. 2024. Symbolic learning enables self-evolving agents. *arXiv preprint arXiv:2406.18532*.

Yuqi Zhu, Shuofei Qiao, Yixin Ou, Shumin Deng, Ningyu Zhang, Shiwei Lyu, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024. Knowagent: Knowledge-augmented planning for llm-based agents. *arXiv preprint arXiv:2403.03101*.

A Related Works

A.1 Agent Planning

Recent studies has shown that in the realm of complex task-solving (Ouyang and Li, 2023; Sun et al., 2023; Liu et al., 2024c, 2025), the capacity for planning and refinement within large models has become increasingly pivotal. It has marked a transition from early Methods like CoT(Wei et al., 2022), Plan and Solve (Wang et al., 2023a), which tackle tasks sequentially, to the sophisticated agentic workflows of today, where model planning is instrumental in addressing a myriad of complex tasks, including question answering (QA) (Mavi et al., 2022), embodied interaction (Yao et al., 2022), tool invocation (Masterman et al., 2024), and long-form text generation (Jiang et al., 2024).

However, initial planning efforts are fraught with deficiencies due to the complexity of environments. When faced with unfamiliar environments, relying solely on human-written task descriptions without interaction with the environment often leads to plans that are misaligned with the actual tasks, or plans that seem reasonable but fail during execution due to a lack of accurate action knowledge. Consequently, there has been a surge in research (Song et al., 2023b; Wang et al., 2023b) focused on refining plans and workflows. These efforts typically leverage direct environmental feedback or design a reward score for end-to-end plan correction, but they often lack a medium for the intermediate processes, which obscures the transparency of the plan refinement process. Moreover, the refinement of plans and the collection of feedback are usually linear and iterative (Qu et al., 2024), resulting in low efficiency and a lack of diversity.

A.2 Knowledge-Augmented Agents

LLMs, as agents interacting with specific environments, often need to provide action signals to these environments (Zhou et al., 2023, 2024; Durante et al., 2024). These action signals can be either restricted or open actions related to the environment (Wang et al., 2024; Li et al., 2024; Hu et al., 2024; Zhang et al., 2024). For instance, they might involve specific movements in embodied task scenarios or the use of various tools like search or OCR in tool invocations. By incorporating these actions, on one hand, the Agent gains the ability to interact with the environment, allowing LLMs to transcend mere textual output (Shridhar et al., 2021; Wang et al., 2022; Zhao et al., 2024b). On the other hand, these external actions endow the Agent with a capability similar to humans using tools, compensating for the inherent limitations of LLMs, such as search tools that can alleviate issues of knowledge hallucination or obsolescence in LLMs (Singh et al., 2024; Zhao et al., 2024c; Shen et al., 2024; Chen, 2023).

Current methods for learning action knowledge are mainly divided into two categories: one involves creating a large amount of synthetic data to construct trajectories for executing actions to train the model (Qiao et al., 2024; Huang et al., 2024b; Zhu et al., 2024), which is costly and has poor generalizability across different tasks; the other relies on prompt engineering (Raman et al., 2022), placing explicit action knowledge about how to plan to execute actions within the prompt, and then using ICL (In-Context Learning) methods to enable the model to learn to invoke these actions. While convenient, these methods can be inaccurate as the artificially constructed planning knowledge may not accurately reflect the true state of the environment, leading to potential biases.

B Setting

B.1 Datasets

ToolBench contains tasks using over 16,000 RapidAPI tools. It assesses a model's ability to plan and execute complex workflows.

HotpotQA is a multi-hop QA dataset with questions requiring multiple steps to answer. We employ Goolge Search as the search engine in the experiment.

B.2 Evaluation

ToolBench: Evaluated using pass rate and win rate. We record planning steps and tool invocations, then submit the trajectory for assessment. Win rate is compared to React's performance. HotpotQA: Evaluated using F1 score, comparing model answers to gold answers (reward 0–1). These datasets and metrics allow us to rigorously validate our approach across varied contexts.

B.3 Baselines

Several strong methods are selected as our baselines, including: ReAct (Yao et al., 2023) which interacts with environment to reason the next step, Self-Refine (Madaan et al., 2023) which uses the feedback from environment to refine the origin prompt, Easy-Tool (Yuan et al., 2024) which

uses llm firstly to refine the tool description and then break down the tasks to complete them, and DRAFT (Qu et al., 2024) to synthesize tasks on a single tool for exploration to learn how to use the tool.

B.4 Experiment Setup

The backend model used in our experiments is Qwen-Long-0916, while the version of GPT-4 is 0613. The token usage in our method is approximately 6-8 million tokens. We configured the width of MCTS to 3 and set the similarity threshold to 0.6. After balancing effectiveness and cost, we synthesized 200 scenes and conducted 15 iterations on them during the experiment.

C Prompt Template

See in Table 3, 4

D Algorithm

See in Algorithmic 1

Algorithm 1 Monte Carlo Tree Search (MCTS) for Action Knowledge Optimization

```
function MCTS(root_node)
    Iteration ← 0
    while Iteration < max_iteration do          ▷ Step 1: Selection with UCB algorithm and num_child < 3
        leaf_node ← SELECT_NODE(root_node)
                                                ▷ Step 2: Expansion
        new_node ← EXPAND(leaf_node)
                                                ▷ Step 3: Simulation
        simulation_result ← SIMULATE(new_node)
                                                ▷ Step 4: Backpropagation
        BACKPROPAGATE(new_node, simulation_result)
        Iteration ← Iteration + 1
    end while
end function
function SELECT_NODE(node)
    while node.is_fully_expanded() do
        node ← CHOOSE_BEST_CHILD(node, exploration_parameter)
    end while
    return node
end function
function EXPAND(node)
    optimization ← CHOOSE_UNTRIED_OPTIMIZATION(node)
    new_node ← APPLY_OPTIMIZATION(node, optimization)
    ADD_CHILD(node, new_node)
    return new_node
end function
function SIMULATE(node)
    optimized_score ← CALCULATE_SCORE(node.current_action_knowledge)
    reward ← optimized_score − father_score
    return reward
end function
function BACKPROPAGATE(node, result)
    while node ≠ None do
        UPDATE_STATISTICS(node, result)
        node ← node.parent
    end while
end function
function CALCULATE_SCORE(action_knowledge)
    return EVALUATE(action_knowledge)
end function
```

Prompt for Tool Description in Action knowledge
Analyze the following tool execution trajectories to improve tool interface documentation. For all trajectories: 1. Identify functional mismatches between original description and actual usage patterns 2. Detect parameter inefficiencies (missing/underutilized fields) 3. Extract implicit requirements from error patterns 4. Generate enhanced documentation with: Clear input specifications (required vs optional) Contextual usage guidelines Error prevention tips Response format expectations Here is an **example**. Now it's your turn to analyze the following tool execution trajectories to improve tool interface documentation. tool_name: **tool_name** original_description: **original_description** trajectory: **trajectory** Please provide your Optimize Description for the tool. Just modify the description part and do not change the parameters description. Make Sure your description is clear and concise.

Table 3: Prompt used for tool document refinement.

Prompt for Workflow in Action knowledge
Analyze the provided interaction trajectory and existing workflow steps to derive a generalized, reusable workflow for similar tool calling tasks. 1. Analyzing error patterns (authentication gaps, deprecated endpoints) and tool dependencies from interaction histories. 2. Extracting implicit requirements (authentication, sorting logic) and mandatory parameters from error responses. 3. Structuring a generic workflow with authentication validation, parameter checks, state management between API calls, and error fallbacks. Here is an **example**. Now it's your turn. Existing Workflow: **workflow** Trajectory: **trajectory** Please provide your Optimize Workflow for the task. And make sure your workflow is clear and concise and no longer than 200 words.

Table 4: Prompt used for workflow generation.

Enhancing Retrieval Systems with Inference-Time Logical Reasoning

Felix Faltings[1]*, **Wei Wei**[2], **Yujia Bao**[2]
[1]MIT, [2]Center for Advanced AI, Accenture
faltings@mit.edu, {wei.h.wei, yujia.bao}@accenture.com

Abstract

Traditional retrieval methods rely on transforming user queries into vector representations and retrieving documents based on cosine similarity within an embedding space. While efficient and scalable, this approach often fails to handle complex queries involving logical constructs such as negations, conjunctions, and disjunctions. In this paper, we propose a novel inference-time logical reasoning framework that explicitly incorporates logical reasoning into the retrieval process. Our method extracts logical reasoning structures from natural language queries and then composes the individual cosine similarity scores to formulate the final document scores. This approach enables the retrieval process to handle complex logical reasoning without compromising computational efficiency. Our results on both synthetic and real-world benchmarks demonstrate that the proposed method consistently outperforms traditional retrieval methods across different models and datasets, significantly improving retrieval performance for complex queries.

1 Introduction

Retrieval systems are integral to many applications, including search engines, question-answering systems, and recommendation platforms (Baeza-Yates et al., 1999; Lewis et al., 2020; Gao et al., 2023). Modern systems operate by transforming user queries into vector representations and retrieving documents based on cosine similarity within an embedding space (Reimers, 2019; Wang et al., 2023; Zhao et al., 2024; Lee et al., 2024). This approach is highly efficient and scalable, as cosine similarity computations are fast and can handle large-scale data. However, the reliance on cosine similarity and static embeddings often limits the system's ability to understand and process complex queries that involve logical constructs such as negations.

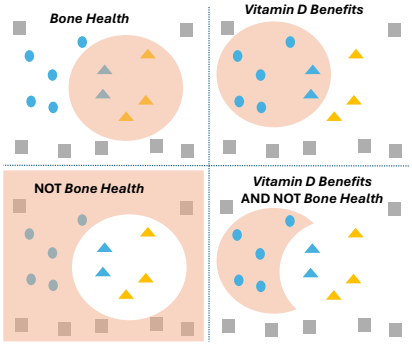


Figure 1: Given a query "What are the benefits of vitamin D, focusing on benefits other than bone health?", we first convert the query into the logical expression "Vitamin D Benefits AND NOT Bone Health". We then calculate the cosine similarity scores for each term (top row) and combine these scores to generate the final results.

Large Language Models (LLMs) have demonstrated remarkable capabilities in inference-time reasoning (Wei et al., 2022; Yao et al., 2024). Recently, (Jiang et al., 2023; Gu et al., 2022; Sun et al., 2023; Luo et al., 2023) have successfully applied LLM's reasoning capability to improve the retrieval performance for knowledge-based question answering, however, the application of inference-time reasoning for general retrieval systems remains relatively unexplored.

We posit that integrating logical reasoning at inference time is equally crucial for retrieval tasks, especially when dealing with complex queries that cannot be accurately represented using simple similarity measures (Meghini et al., 1993; Ounis and Paşca, 1998). Consider a query like "What are the benefits of vitamin D, focusing on benefits other than bone health?" A traditional retrieval system, relying solely on cosine similarity, may struggle to exclude documents related to bone health due to the embedding's inability to represent the negation effectively. The system tends to retrieve documents that are globally similar to the query, failing to account for specific logical instructions, such as

*Work done during an internship at Accenture.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 449–463
July 27 - August 1, 2025

exclusions or combinations of concepts.

To address this limitation, we propose a novel inference-time reasoning framework for retrieval systems that explicitly incorporates logical reasoning into the retrieval process. Our approach involves three key steps. First, **Logical Query Transformation**: We utilize an LLM to parse and rewrite the natural language query into a logical form, such as "A AND B OR NOT C," where A, B, and C represent different semantic components of the query. Second, **Term Embedding and Similarity Computation**: Each term identified in the logical form (A, B, C, etc.) is individually encoded into the embedding space. We compute the cosine similarity between each term's embedding and the embeddings of documents in the corpus. Third, **Score Composition Based on Logical Relations**: We combine the similarity scores of each term in accordance with their logical relations. Our approach adds minimal overhead since the embedding can be performed in parallel.

We validate our approach through comprehensive experiments. We first create synthetic datasets with queries of varying logical complexity to test the limitations of existing retrieval algorithms. Our findings indicate that as the number of logical terms increases, the performance of traditional retrieval methods degrades significantly, while our method better maintains performance, demonstrating robustness against query complexity. We also evaluated our algorithm on three real-world datasets: NFCorpus (Boteva et al., 2016), SciFact (Wadden et al., 2020) and ArguAna (Thakur et al., 2021). Specifically, we augmented these three datasets with natural language queries that target compositional reasoning. We tested our method using four commonly used embedding models. The results show that our approach consistently outperforms baseline methods across all models and datasets, confirming its effectiveness in practical scenarios.

2 Method

Given a natural language query, *"What are the beneifts of vitamin D, focusing on benefits other than bone health?"*, we first transform it into a logical expression using a pre-trained large language model (Dubey et al., 2024), *"Vitamin D Benefits" AND NOT "Bone Health"*, where the terms in quotes can be any string of text. Given a document, these queries can be seen as logical expressions, which we evaluate in a fuzzy way (Novák et al., 2012), using the scores from a dense retrieval model to assign truth values to each clause. The fuzzy evaluation of the expression then gives a composite retrieval score for the given document. In the following sections, we present the concrete details of our method, starting with the syntax of the logical queries, followed by the retrieval semantics.

2.1 Query Syntax

Queries are made up of terms–which can be any string of text–combined using operators. We allow three operators, AND, OR, and NOT. Formally, the syntax of the language is described by the following simple grammar,

$$\begin{aligned} T &\rightarrow U \text{ OR } U \mid U \\ U &\rightarrow V \text{ AND } V \mid V \\ V &\rightarrow \text{ NOT } W \mid W \\ W &\rightarrow \textit{string} \mid (T) \end{aligned}$$

where the use of distinguished non-terminals, T, U, V, and W enforces an operator priority, NOT $\succ$ AND $\succ$ OR , which is itself overridden by parentheses.

2.2 Query Semantics

For each term t_j in a query, and each document D_i in a corpus, we compute a score s_{ji} using the dense retrieval model. Usually, this is the cosine similarity between the embedding vectors of the term and document. The semantics of the query then tell us how to combine the scores s_{ji} into a single score s_i which we can use for retrieval.

Consider a query of the form,

$$(t_1 \text{ OR } t_2) \text{ AND NOT } t_3.$$

Then, for document D_i, if s_{1i}, s_{2i}, and s_{3i} are the scores obtained from the dense retrieval model, we take the composed retrieval score to be,

$$s_i = \text{OP}_{AND}(\text{OP}_{OR}(s_{1i}, s_{2i}), \text{OP}_{NOT}(s_{3i})),$$

where OP_{AND}, OP_{OR} and OP_{NOT} are functions that define how scores should be combined depending on the query operator. We detail our choice of operators in the next section.

In general, each query can be parsed using the grammar described above, resulting in a parse tree, which is directly translated into a tree of operations acting on the scores s_{1i}, s_{2i}, and s_{3i} as shown in Fig. 2. A more complicated example is given in the appendix. This tells us how to compute the final retrieval score s_i. More formally, this could be written as an attribute grammar (Knuth, 2005).

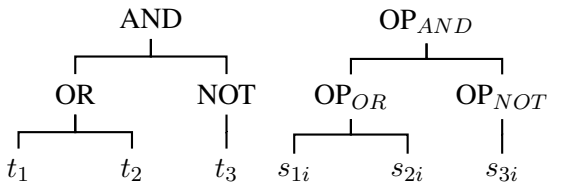


Figure 2: Example parse tree (left) and corresponding graph of operations (right).

2.3 Score Operations

The choice of the operators OP_{AND}, OP_{OR}, OP_{NOT} should reflect the logical semantics of the query. For example, for the conjunction t_1 AND t_2, with scores s_{1i} and s_{2i}, the composite score should be low if *any* of the two scores is low. In this work we consider three approaches,

1. First, we take the perspective of fuzzy logic, where we view the retrieval scores as fuzzy set memberships. The fuzzy membership to a union of sets, corresponding to the logical OR operation, is then taken to be the maximum of the individual fuzzy memberships, and the membership to an intersection, corresponding to the AND operation, is given by the minimum (Fox and Sharat, 1986).

2. Second, we also choose operators based on probability theory. The probability of two independent events occurring simultaneously is $P(A \cap B) = P(A)P(B)$, the probability of the union of two events is upper bounded by $P(A \cup B) \leq P(A) + P(B)$ and the probability of the complement of an event is $P(\neg A) = 1 - P(A)$. Motivated by this we use the arithmetic operations $\times$, $+$ and $1 - \cdot$ for AND, OR and NOT respectively.

3. Finally, we also use heuristically motivated operators such as addition for AND, so that a document that scores highly for individual queries will score highly for their conjunction; and $1/\cdot$ for NOT, so that a document that scores highly for a query will score low for its negation.

In summary, the full set of operator choices is,

$$\text{OP}_{AND}(x, y) = x * y \,|\, x + y \,|\, \min(x, y)$$
$$\text{OP}_{OR}(x, y) = x + y \,|\, \max(x, y)$$
$$\text{OP}_{NOT}(x, y) = 1 - x \,|\, 1/x$$

We evaluate all combinations of these operators in our experiments. Our default choice is $\text{OP}_{AND}(x, y) = x * y$, $\text{OP}_{OR}(x, y) = x + y$, and $\text{OP}_{NOT}(x, y) = (1 - x)$, which we find to work best.

3 Results

We start by validating the performance of the logical retrieval system itself on synthetic data. Next, we assess the system's utility for retrieval on real data. In all our experiments, we evaluate using four base embedding models: Nvidia's `NV-Embed-V1` (Lee et al., 2024), Mistral's `nv-embedqa-mistral-7b-v2` and OpenAI's `text-embedding-v3-large` and `text-embedding-v3-small`.

3.1 Synthetic Data

Three Term Queries We first evaluate performance on all possible queries formed of three terms using synthetic data. This gives 32 "templates", such as,

$$t_1 \text{ AND } t_2 \text{ OR NOT } t_3$$

The three placeholders t_1, t_2, and t_3 are filled in by terms. For each possible template, we generate 100 queries by filling in the placeholders with random topics from a set generated by Llama3-70b. For each query we then generate documents that match and don't match the query with Llama3-70b. For example, for the query,

"dog" AND "cat" AND "mouse",

we generate one document that matches, which is related to all three terms, and three documents that don't match, which are related to all but one of the terms. See Appendix H for more details.

The results are presented in Table 1. We report the standard nDCG@10 in all our results. We show the results when passing the query directly to the retrieval model (base) vs. composing the retrieval model scores for each term (logical). For reference, the performance when using random scores is around 0.7. We see that logical retrieval outperforms the baseline, with the most gains coming from queries with negations. We did not see large differences between embedding models. See Table 7 in the appendix for a breakdown.

Scaling Number of Queries We look at performance as the number of terms in the queries scales, focusing this time solely on queries consisting of OR or AND operators. For example,

"dog" AND "mouse" AND ... AND "cat"

	Number of negations			
	0	1	2	3
Base	0.95	0.77	0.65	0.52
ITLR	**0.99**	**0.97**	**0.96**	**1.00**

Table 1: nDCG@10 Results on synthetic data. Dense and logical retrieval systems were evaluated on synthetically generated test cases for all 32 possible logical queries with three terms. We show results broken down by the number of negations in the queries.

		OP_{NOT}	
OP_{AND}	OP_{OR}	$1-x$	$1/x$
$\min(x,y)$	$\max(x,y)$	0.86	0.86
	$x+y$	0.92	0.87
$x*y$	$\max(x,y)$	0.90	0.89
	$x+y$	**0.97**	0.90
$x+y$	$\max(x,y)$	0.91	0.81
	$x+y$	**0.97**	0.82

Table 2: nDCG@10 results for logical retrieval on the same data from Table 1 for different choices of operators, using `NV-Embed-V1`. Each entry represents a choice for each of the three operators.

We generate data in the same way as before. Our results are presented in Fig. 3. We see that the gains from logical retrieval increase as the number of terms increases. This is more pronounced for the AND queries than the OR queries, likely since each AND query has a single positive match whereas each OR query has many matches.

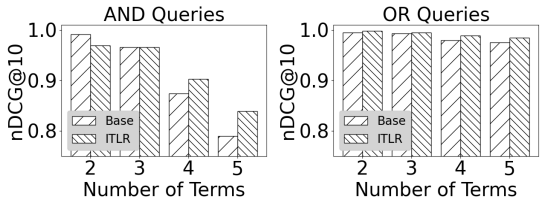


Figure 3: Performance as the number of terms scales. Baseline dense retrieval and logical retrieval were evaluated on queries connected by AND and OR clauses, with increasing number of clauses.

3.2 Operator Combinations

We tested all combinations of operators OP_{AND}, OP_{OR} and OP_{NOT} proposed in section 2.3 on the three term query data using the `NV-Embed-V1` embedding model. We present the results in Table 2. We see that our default choice works best, although the alternative using $\text{OP}_{AND}(x,y) = x+y$ works equally as well. Note that the common choice made in fuzzy logic of $\text{OP}_{AND}(x,y) = \min(x,y)$ and $\text{OP}_{OR}(x,y) = \max(x,y)$ performs quite poorly.

3.3 Real Data

Our previous experiments consider very small corpora constructed specifically for each query. We now turn to real datasets. In order to ensure the queries have sufficient compositionality, we generate queries using Llama3-70b. As in our synthetic experiments, we create 3-term logical queries from templates, filled in with topics extracted from the dataset. We create 30 queries per template, resulting in 960 total queries. We ask Llama3-70b to turn these queries into natural language questions and throw away the original queries. We also use Llama3-70b to label the relevance of each document in the corpus to each of the questions. In the appendix we give the prompts that were used and show through examples that the generated queries are realistic.

	NFcorpus	**SciFact**	**ArguAna**	**Legal**
`NV-Embed-V1`:				
Baseline	0.56	0.51	0.51	0.46
BRIGHT	0.67	0.59	0.58	0.52
ITLR	**0.74**	**0.64**	0.64	0.59
`text-embedding-v3-large`:				
Base	0.63	0.59	0.63	0.56
BRIGHT	0.70	0.63	0.66	0.60
ITLR	0.73	**0.64**	**0.67**	**0.66**
`text-embedding-v3-small`:				
Base	0.56	0.50	0.55	0.50
BRIGHT	0.68	0.59	0.65	0.57
ITLR	0.67	0.54	0.63	0.61
`nv-embedqa-mistral-7b-v2`:				
Base	0.54	0.50	0.40	0.41
BRIGHT	0.48	0.39	0.29	0.27
ITLR	0.67	0.61	0.59	0.42

Table 3: nDCG@10 Results on real data. For each dataset taken from BEIR (Thakur et al., 2021), compositional questions were generated using Llama3-70b. We show results for three embedding models and three methods. Legal short for LegalBenchCorporateLobbying.

	Number of negations			
	0	**1**	**2**	**3**
Base	**0.81**	0.60	0.51	0.36
Reasoning	**0.81**	0.68	0.64	0.56
Logical	0.76	**0.71**	**0.76**	**0.73**

Table 4: Breakdown of Table 3 for `NV-Embed-V1` on NFCorpus by number of negations.

We compare three methods. The **Baseline** feeds the question to the dense retrieval model. The **BRIGHT** baseline first asks Llama3-70b to reason about the question and feeds the reasoning trace to the retrieval model. This is the method used in (Su et al., 2024). Finally, our **ITLR** method asks Llama3-70b to generate a logical query from the question, which is fed to our logical retrieval system. See Sec. 3.4 for human evaluation results

Base Dataset	Corpus Size	Gen. Queries
NFCorpus	3,633	960
ArguAna	8,674	960
SciFact	5,183	960
Legal	340	960

Table 5: **Dataset statistics** Legal short for LegalBenchCorporateLobbying.

	NFCorpus	SciFact	ArguAna
Accuracy	87.29	86.25	85.1

Table 6: Accuracy of question transformation into logical queries.

showing that Llama3-70b is able to successfully formulate logical queries.

We report nDCG@10 results in Table 3, for four datasets derived respectively from the NFCorpus, SciFact, ArguAna and LegalBenchCorporateLobbying tasks, accessed through MTEB (Muennighoff et al., 2022). Table 5 gives the statistics of the datasets used in our experiments.

We see that ITLR achieves the best performance overall, beating all baselines in a majority of cases. This performance gap becomes larger with more negations, as seen in Table 4. The lack in improvement without negations may be due to the use of three term queries. Since simpler queries are likely better represented in the training distribution of the retrieval models, they are easy enough to process. In our synthetic experiments (Figure 3) we show that, as the number of terms increases, ITLR outperforms baselines including on cases without negations.

3.4 Accuracy of Query Transformation

Because our system is reliant on transforming user queries from natural language into our logical syntax, we investigated the accuracy of Llama3-70b on this query transformation task. For all the questions in three of our real datasets–NFCropus, SciFact and ArguaAna–we asked human annotators to assess whether or not the logical query generated by the LLM accurately reflected the original question. As can be seen in Table 6, the generated logical queries tend to capture the original question quite well, with accuracies of 85% to 87%. In appendix B we also examined the queries labelled as inaccurate and found that in the cases we considered the errors tended to be minor.

4 Conclusion

In this paper, we propose an inference-time logical reasoning framework that addresses the limitations of traditional retrieval methods in managing complex queries with logical constructs. The framework is highly efficient, enabling concurrent computation of retrieval scores for each term. By integrating logical reasoning directly into the retrieval process, our framework consistently outperforms traditional methods on both synthetic and real-world benchmarks, demonstrating particular strength in handling queries with a higher frequency of negations and AND operations.

5 Limitations

The logical retrieval system we presented in this paper presents some limitations, as it still underperforms on queries without any negations. We identify some concrete problems that could be addressed in future work. First, in most scenarios, the queries to retrieval systems, such as questions from users, are not given as logical formulas. It is also unreasonable to expect users to write logical formulas on their own. Hence, the system is reliant on reformulating queries into logical queries. While we used a simple prompt to achieve this, it is possible that better performance could be obtained by finetuning a reformulation model. Second, we observed in preliminary experiments that the performance of the system can be improved by calibrating the scores of the underlying retrieval model. For example, when processing AND queries, some terms may receive generally higher retrieval scores than others, biasing retrieval towards documents that match those terms but not the others. We did not find any simple methods to calibrate the scores, but this could be accomplished when training the retrieval model, or by training a calibration model on a large dataset.

Ethical Considerations We do not foresee any immediate risks of our work as we are not releasing any major new artifacts, such as pre-trained models, which could be used in adverse ways. Retrieval systems are limiting points in applications such as retrieval augmented generation (RAG) since downstream answers are generated based on the documents provided by the retrieval system. Poor retrieval systems may skew the information retrieved from corpora, and thus improving the faithfulness of retrieval systems may be broadly beneficial.

References

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval*, volume 463. ACM press New York.

Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Edward A Fox and S Sharat. 1986. A comparison of two methods for soft boolean interpretation in information retrieval. Technical report, TR-86-1. Virginia Tech. Department of Computer Science.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Yu Gu, Xiang Deng, and Yu Su. 2022. Don't generate, discriminate: A proposal for grounding language models to real-world environments. *arXiv preprint arXiv:2212.09736*.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data. *arXiv preprint arXiv:2305.09645*.

Donald E Knuth. 2005. The genesis of attribute grammars. In *Attribute Grammars and their Applications: International Conference WAGA Paris, France, September 19–21, 1990 Proceedings*, pages 1–12. Springer.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Haoran Luo, Zichen Tang, Shiyao Peng, Yikai Guo, Wentai Zhang, Chenghao Ma, Guanting Dong, Meina Song, Wei Lin, Yifan Zhu, et al. 2023. Chatkbqa: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models. *arXiv preprint arXiv:2310.08975*.

Carlo Meghini, Fabrizio Sebastiani, Umberto Straccia, and Costantino Thanos. 1993. A model of information retrieval based on a terminological logic. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 298–307.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Vilém Novák, Irina Perfilieva, and Jiri Mockor. 2012. *Mathematical principles of fuzzy logic*, volume 517. Springer Science & Business Media.

Iadh Ounis and Marius Paşca. 1998. Relief: Combining expressiveness and rapidity into a single system. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval*, pages 266–274.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S Siegel, Michael Tang, et al. 2024. Bright: A realistic and challenging benchmark for reasoning-intensive retrieval. *arXiv preprint arXiv:2407.12883*.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model with knowledge graph. *arXiv preprint arXiv:2307.07697*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60.

Embedding Model	**Base**	**ITLR**
`NV-Embed-V1`	0.74	**0.97**
`text-embedding-v3-large`	0.71	**0.97**
`text-embedding-v3-small`	0.71	**0.97**

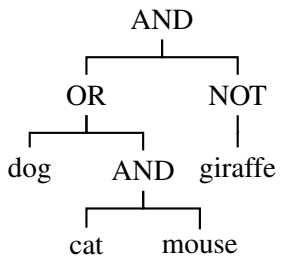
Table 7: nDCG@10 results on synthetic data, broken down by embedding model.

A Additional Parsing Example

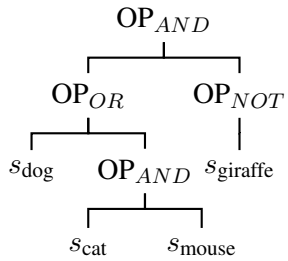
Consider the query,

(“dog" OR “cat" AND “mouse") AND NOT “giraffe"

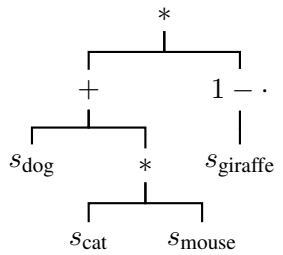
The corresponding parse tree will be,



And the computational graph for combining the scores is,



With our actual choice of operators this looks like,



Written out as a formula this gives the final retrieval score as

$$s = (s_{\text{dog}} + s_{\text{cat}} * s_{\text{mouse}}) * (1 - s_{\text{giraffe}}).$$

B Additional Results

Breakdown by Embedding Model Table 7 breaks down the synthetic data results by embedding model. We see that there is little difference between the embedding models we considered.

Accuracy of Query Transformation We investigated how accurate Llama3-70b was at transforming natural language questions into logical queries. Below we give an example of inaccurate queries from the NFCorpus, SciFact and ArguAna datasets respectively.

- **Example 1** Natural language question:

What are some studies on the health benefits of anthocyanins that are not specifically focused on cancer prevention or targeted towards men?

LLM generated query:

"Health benefits of anthocyanins" AND (NOT "cancer prevention" OR NOT "targeted towards men")

Here, the original question is ambiguous about which studies should be excluded.

- **Example 2** Natural language question:

  What are some recent research topics in the fields of neurology and oncology?

  LLM generated query:

  "Recent research in neurology" OR "Current research topics in neurology" AND "Recent advancements in oncology" OR "Current studies in oncology"

  Here the AND should have been an OR .

- **Example 3** Natural language question:

  What are some news articles that are not about agricultural development in Africa or Brexit, or are about a topic unrelated to peacekeeping?

  LLM generated query:

  "News articles" AND NOT ("Agricultural development in Africa" OR "Brexit" OR "Peacekeeping")

  Here, "Peacekeeping" should be outside the NOT clause.

C Data Examples

Below we provide examples of queries from our NFCorpus dataset.

- **Example 1**

  What are the risk factors for pancreatic cancer, excluding those related to MRSA in swine farms, or what are the benefits of cruciferous compounds in cancer prevention?

- **Example 2**

  What are the health benefits and risks of a vegetarian diet that does not include dairy products, and are there any natural alternatives to dairy that can provide similar nutritional value?

- **Example 3**

  What are some ways to prevent cancer through diet, excluding the effects of xenohormesis mechanisms, and specifically considering the potential benefits of cherries or other foods rich in phenolic compounds?

- **Example 4**

  What are the dietary factors that can help prevent cancer, excluding those related to polycyclic aromatic hydrocarbons?

- **Example 5**

  What are the effects of bioactive compounds on colon or prostate cancer, excluding studies on their mechanisms of action?

Below, we give the corresponding logical queries generated by the LLM,

- **Example 1**

  ("Risk factors for pancreatic cancer" AND NOT "MRSA in swine farms") OR "Benefits of cruciferous compounds in cancer prevention"

- **Example 2**

  "Health benefits of a vegetarian diet without dairy products" AND "Risks of a vegetarian diet without dairy products" AND ("Natural alternatives to dairy products" OR "Plant-based alternatives to dairy products") AND ("Nutritional value of dairy products" OR "Nutritional benefits of dairy alternatives")

- **Example 3**

  "Dietary prevention of cancer" AND NOT "xenohormesis" AND ("cherries" OR "foods rich in phenolic compounds")

- **Example 4**

  "Dietary factors that help prevent cancer" AND NOT "Polycyclic aromatic hydrocarbons"

- **Example 5**

  "Effects of bioactive compounds on colon cancer" OR "Effects of bioactive compounds on prostate cancer" AND NOT "Mechanisms of action of bioactive compounds"

D Reformulation Prompts

Here we give the prompts used for query reformulation in our reasoning methods from Sec. 3.3.

BRIGHT Reasoning prompt

```
Here is a user query:

\{question\}

(1) Identify the essential question in the query.
(2) Think step by step to reason about what should be included in the relevant
↪  documents.
(3) Draft an answer.
```

Logical Formula Prompt

```
I have a document retrieval system that processes logical queries.
These queries can be of the form,
"term1" AND "term2" OR "term3" AND NOT "term4"

The meaning of the operators AND, OR and NOT should be obvious:
- AND means the retrieved document should be related to both terms
- OR means the retrieved document can be related to either term
- NOT means the retrieved document should not be related to the given term

Given a natural language question from a user, I want to use the retrieval system to
↪  gather documents that contain information relevant to the user's question.
I need you to create suitable logical query to the retrieval system.
Remember that each of the individual terms can be a keyword, a phrase, a sentence,
↪  or even
```

	Number of negations			
	0	1	2	3
Base	0.95	0.77	0.65	0.52
ITLR	0.99	0.97	0.96	**1.00**
Calibrated ITLR	**1.00**	**0.98**	**0.97**	**1.00**

Table 8: Expanded version of Table 1 including calibration.

```
a whole document. So don't limit yourselves to keywords.
For example, the following question,
"What is the impact of eating fresh oranges on pancreatic cancer risk,
and its relationship to stage II diabetes"
Could be answered with the query,
"Impact of fresh orange consumption on pancreatic cancer risk" AND
"What is the relationship of eating fresh oranges to stage II diabetes?"
A query that only used keywords, such as
"oranges" AND "consumption" AND "pancreatic cancer" AND "stage II diabetes"
would lose much of the meaning of the original question! It's not clear if
↪  consumption relates
to oranges, so a document that talks about consuming figs, and peeling oranges
would match this query!

Here is the user's question,
\{question\}

Can you come up with a suitable logical query to the retrieval system? Only include
↪  the query
in your answer.
```

E Calibration

We also experimented with calibrating the scores from the retrieval models before combining them in ITLR. On the synthetic data, for a given embedding model and for each term, we generated 20 positive and 20 negative documents using Llama3-70b. This gave us a dataset of documents $x_1, \ldots, x_N$, with embedded cosine similarities $s_1, \ldots, s_N$ and labels $y_1, \ldots, y_N \in \{0, 1\}$. We then fit a simple calibration model,

$$\hat{y}_i = \sigma((s_i - \tau) * \lambda),$$

using gradient descent. This calibration offered some improvements on our synthetic dataset, as can be seen in Table 8, which is an expanded version of Table 1 with calibration.

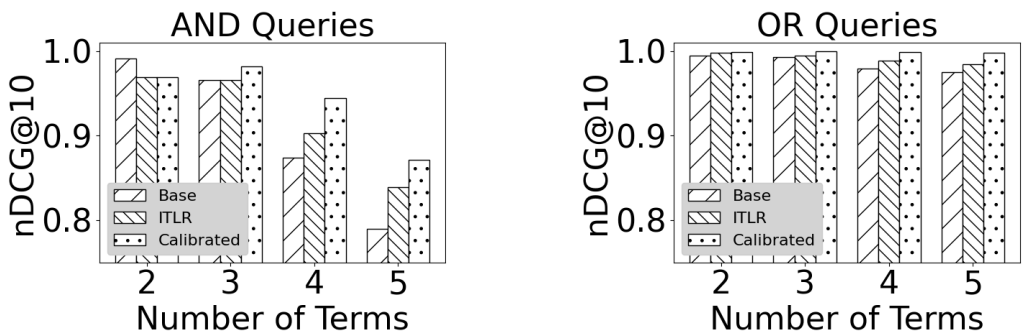


Figure 4: Expanded version of Fig. 3 including calibration.

Model	Number of Model Parameters	Access	License
Llama3-70b	$70 * 10^9$	API	cc-by-nc-4.0
NV-Embed-V1	$7.85 * 10^9$	API	llama3
text-embedding-3-large	Undisclosed	API	commercial
text-embedding-3-small	Undisclosed	API	commercial

Table 9: Dataset statistics

F Additional Data Details

The datasets were accessed through MTEB (Muennighoff et al., 2022) under an Apache 2.0 license. These datasets all contain English text.

G Model Details

Table 9 gives details on the embedding models and LLMs used in our experiments, including parameter counts and how they were accessed.

H Synthetic Data Generation

Consider a query,

"mouse" OR "dog" AND NOT "cat".

Any document that matches this query can be categorized by the individual terms of the query that it does or does not match. For example, for this query, there are two categories,

1. Matches mouse but not cat
2. Matches dog but not cat

Any of these categories is a list of terms that the document matches, and a list of terms that it doesn't match. This gives an easy way to generate matching documents using Llama3-70b by providing terms that should or shouldn't be matched.

There are many documents that don't match the query, but we want to evaluate against challenging negatives. We can create these by taking one of the categories above and swapping a condition. For example, a positive document will match "mouse" but not "cat". A hard negative could match "mouse" *and* "cat". In general, given a list of terms that the document should or shouldn't match, we simply move one term to the opposite category. So a hard negative will match all the terms it should to match the query, except one. Or it will avoid all the terms it should, except one.

Thus, to generate documents for our synthetic data, we first enumerate all the combinations that positive documents should or shouldn't match. We then create all the hard negatives by swapping one of the terms. For each set of terms to match or not match, we create one document. For three term queries, this will result in at most three positive documents. We also keep up two three negative documents. Hence for each query, we generate up to six documents.

I Synthetic Queries

We found that the queries in original datasets used in our experiments are overly simple. For example, in the NFCorpus dataset the queries are the titles of web articles such as "Philippines". To better evaluate the retrieval performance under complex user questions, we created a set of queries based on topics found in the corpus and then used an LLM to evaluate how relevant each passage was to each query. The passages are completely unchanged. This gives us natural and challenging queries with a better labelling of relevant passages, while retaining the complexity of a real world document corpus.

To generate the queries, we extract topics from random documents in the corpus. We then create logical queries using the extracted topics. Finally, we transform these into natural language questions. The Llama3-70b prompts we used are given below.

Topic Extraction Prompt

```
You will be given a document. You need to extract all the salient topics from it.
The topics should range from general to specific. Here is the document:
{format_doc(doc)}

Please give the salient topics as a list with one topic per line.
Don't include anything else in your answer.
Sort the topics from most general to most specific.
```

Query to Question Prompt

```
I have a document retrieval system that processes logical queries.
These queries can be of the form,
"term1" AND "term2" OR "term3" AND NOT "term4"

The meaning of the operators AND, OR and NOT should be obvious:
- AND means the retrieved document should be related to both terms
- OR means the retrieved document can be related to either term
- NOT means the retrieved document should not be related to the given term

I want to evaluate the performance of a human user to use this retrieval system.
Given a natural language question, the user needs to come up with a logical query
↪  that will best
retrieve relevant documents.
In order to make a dataset for evaluation, I want to operate in reverse. I have
↪  collected
many logical queries, and I would like to come up with a corresponding natural
↪  language question.
Then I can give the question to a user and see how well the recover they original
↪  query.

So, given the following logical query, can you come up with such a natural language
↪  question?
Here's the query,
{query}

What question would you come up with? Only include the question in your answer.
```

J Human Annotations

Human annotators tasked with evaluating the LLM generated queries were paid a fair wage of $25 an hour. They were given the following instructions

```
Instruction for Human Annotator: Logical Expression Validation
Task Overview
You will be given a natural language question and a corresponding logical
↪  expression generated by an LLM (Large Language Model). Your task is to
↪  determine whether the logical expression accurately represents the intended
↪  meaning of the question.
A correct logical expression should:
- Capture the key intent of the question.
- Properly reflect any exclusions, inclusions, or constraints mentioned.
- Maintain the logical relationships between elements.
```

Evaluation Criteria
1. Accuracy - Does the logical expression correctly interpret the intent of the question?
2. Completeness - Are all relevant aspects of the question included in the logical expression?
3. Exclusions - If the question explicitly excludes something, does the logical expression handle this correctly?
4. Logical Structure - Are the AND, OR, and NOT operators used correctly to reflect the relationships in the question?
If the logical expression is correct, mark it as valid. If incorrect, mark it as invalid and provide an explanation of the error.
Examples
Positive Examples (Correct Expressions)
Example 1:
- Natural Language Question: ``How does vitamin D benefit your health? I already know about bone health, so I want to know other benefits.''
- Parsed Logical Expression: health benefits of vitamin D AND NOT bone health
- Explanation: The logical expression correctly retrieves information about vitamin D's health benefits while excluding bone health, as specified in the question.
Example 2:
- Natural Language Question: ``What are some movies directed by Christopher Nolan, excluding superhero films?"
- Parsed Logical Expression: movies directed by Christopher Nolan AND NOT superhero films
- Explanation: The logical expression correctly filters out superhero films while still retrieving Nolan's other movies.
Example 3:
- Natural Language Question: ``Which laptops have at least 16GB RAM and either an Intel i7 or AMD Ryzen 7 processor?"
- Parsed Logical Expression: laptops AND 16GB RAM AND (Intel i7 OR AMD Ryzen 7)
- Explanation: The expression correctly captures the requirement of 16GB RAM and allows either processor type, as intended.
Negative Examples (Incorrect Expressions)
Example 4:
- Natural Language Question: ``How does vitamin D benefit your health? I already know about bone health, so I want to know other benefits."
- Parsed Logical Expression: health benefits of vitamin D OR NOT bone health
- Error: The use of OR NOT instead of AND NOT changes the meaning. The expression may return results that are completely unrelated to vitamin D.
Example 5:
- Natural Language Question: ``What are some movies directed by Christopher Nolan, excluding superhero films?"
- Parsed Logical Expression: movies directed by Christopher Nolan OR NOT superhero films
- Error: The OR NOT operator incorrectly allows movies that aren't superhero films but might not be directed by Nolan, which is not what the question asks.
Example 6:
- Natural Language Question: ``Which laptops have at least 16GB RAM and either an Intel i7 or AMD Ryzen 7 processor?"
- Parsed Logical Expression: laptops AND (16GB RAM OR Intel i7 OR AMD Ryzen 7)

- Error: The use of OR within the parentheses makes it possible for laptops with only Intel i7 or AMD Ryzen 7 (but less than 16GB RAM) to be included, which is incorrect.

Final Notes

- Pay close attention to negations (NOT). Misplacing them can completely alter the meaning.
- Ensure correct grouping with parentheses. Ambiguities in logic can lead to unintended results.
- Rephrase the natural language question in a structured way before checking the logical expression.

Your accuracy in annotation ensures that the model correctly understands and processes logical constraints in natural language.

Using Subtext to Enhance Generative IDRR

Zhipang Wang, Yu Hong*, **Weihao Sun, Guodong Zhou**
School of Computer Science and Technology, Soochow University, Suzhou, China
{zhipangwang, tianxianer, whsun16}@gmail.com; gdzhou@suda.edu.cn

Abstract

Implicit Discourse Relation Recognition (abbr., IDRR) is a NLP task of classifying argument pairs into different types of semantic relations. Arguments contain subtexts, some of which are beneficial to the perception of semantic relations. However, subtexts are connotative. The neural IDRR model fails to be aware of them without being given pertinent prompts. In this paper, we leverage LLaMA to generate subtexts for argument pairs, and verify the effectiveness of subtext-based IDRR. We construct an IDRR baseline using the decoder-only backbone LLaMA, and enhance it with subtext-aware relation reasoning. A confidence-diagnosed dual-channel network is used for collaboration between in-subtext and out-of-subtext IDRR. We experiment on PDTB-2.0 and PDTB-3.0 for both the main-level and secondary-level relation taxonomies. The test results show that our approach yields substantial improvements compared to the baseline, and achieves higher $F1$-scores on both benchmarks than the previous decoder-only IDRR models. We make the source codes and data publicly available.[1]

1 Introduction

IDRR determines the semantic relation between arguments when the in-between connective is absent (Prasad et al., 2008). For example, it outputs the relation "*Concession*" for the arguments $Arg1$ and $Arg2$ in 1), where the possible connective "*however*" is not given in the source text:

1) **Arg1**: *The new rate will be payable Feb. 15.*
   Arg2: *A record date hasn't been set.*
   Relation: Concession

Encoder-only language models such as RoBERTa (Long and Webber, 2022; Wu et al., 2023; Cai et al., 2024) and XLNet (Jiang et al., 2024) have been used for IDRR, where multi-class relation classification is conducted by linear layers with Softmax. Meanwhile, both T5 (Jiang et al., 2021; Chan et al., 2023) and the decoder-only Large Language Mdoels (LLMs) like GPT-3.5 (Chan et al., 2024) and GPT-4 (Yung et al., 2024) have also been verified for IDRR, where relations are properly generated conditioned on prompts and/or CoT. Significant improvements are reported in these arts.

Subtext hasn't been considered in the study of IDRR. Though, it is potentially useful for enhancing the IDRR models. A subtext is characterized by the metaphorical meaning hidden in the arguments. For example, the subtext of the two arguments in 1), most probably, is "*the rate should be recorded earlier though it hasn't been*". Such a subtext is more explicit or even straight in revealing the *Concessive* relation. Accordingly, we suggest that subtext can be used as a crucial evidence for enhancing the perception of implicit relation.

In this paper, we explore the method of applying subtexts, and systematically investigate the effectiveness upon LLM-based generative IDRR. The effort we made is to provide a preliminary study and stimulate innovative researches in subtext-based IDRR enhancement. Specifically, our contributions are summarized as follows:

- We first propose a Subtext-based Confidence-diagnosed Dual-channel Network (SCDN) for IDRR. In SCDN, subtext is generated by LLM. Confidence comparison is conducted to reconcile in-subtext and out-of-subtext IDRR.
- We verify the effectiveness of SCDN on the benchmarks PDTB-2.0 and 3.0 (Webber et al., 2019) . We report varied influences cased by the settings of prompting, confidence diagnosis, subtext generation and augmentation.

*Corresponding author
[1] https://github.com/ZpWang-AI/IDRR_Subtext.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 464–473
July 27 - August 1, 2025

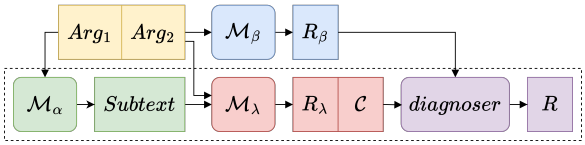


Figure 1: Architecture of SCDN.

2 Approach

Figure 1 shows the architecture of SCDN, which is constructed with three LLMs $\mathcal{M}_\alpha$, $\mathcal{M}_\beta$ and $\mathcal{M}_\lambda$. $\mathcal{M}_\alpha$ serves to generate the subtext for the given arguments. $\mathcal{M}_\beta$ takes the arguments as input and uses them as the only reliance for relation reasoning. $\mathcal{M}_\lambda$ combines the generated subtext and arguments, and infers the relation according to all of them. A probabilistic diagnosis model (*diagnoser*) is used to reconcile the decisions from $\mathcal{M}_\beta$ and $\mathcal{M}_\lambda$ based on confidence estimation.

2.1 Subtext and Relation Generators

Our baseline is the generator $\mathcal{M}_\beta$ which performs out-of-subtext IDRR. We prompt it by Question Answering (QA). Given the arguments $\dot{A}$ and $\ddot{A}$, we combine them with a question Q_β of "*what is the relation between arguments*": I_β=[{$\dot{A}$,$\ddot{A}$}; Q_β]. We feed I_β into $\mathcal{M}_\beta$ to generate a relation label.

To fulfill the in-subtext IDRR, we construct a subtext generator $\mathcal{M}_\alpha$. Its input is formed by the arguments and a prompting question Q_α of "*what is the implicit meaning*": I_α=[{$\dot{A}$,$\ddot{A}$}; Q_α]. There isn't any constraint applied to subtext generation (i.e., $\mathcal{M}_\alpha(I_\alpha)$) such as the length of subtext. Further, we build the generator $\mathcal{M}_\lambda$ to perform in-subtext IDRR. It uses both subtext and arguments as input, and combines them with a multi-choice question Q_λ: I_α=[{$\dot{A}$,$\ddot{A}$}; {S}; Q_λ]. The question Q_λ is designed as "*what is the relation between arguments given subtext*", which allows $\mathcal{M}_\lambda$ to generate a relation label in the manner of multi-choice QA (Yung et al., 2024) as follows.

2) Q_λ: What is the relation of $\dot{A}$ and $\ddot{A}$ given S?
A. *Contingency* B. *Expansion*
C. *Temporality* D. *Comparison*

In our experiments, we uniformly use LLaMA3-8B-Instruct (Dubey et al., 2024) to construct the generators $\mathcal{M}_\alpha$, $\mathcal{M}_\beta$ and $\mathcal{M}_\lambda$. Due to the zero-resource situation that there isn't any ground-truth subtext provided in PDTB-2.0 and 3.0, we train the subtext generator $\mathcal{M}_\alpha$ by teacher-student knowledge distillation (Hu et al., 2023). GPT-3.5-turbo (Brown et al., 2020) is used as the teacher.

2.2 Confidence Diagnoser

It is unavoidable that the subtext-based generator encounters two problems, including 1) arguments inherently don't contain a subtext, and 2) the generated subtext is unqualified. To relieve the problems, we use a diagnoser to reconcile $\mathcal{M}_\beta$ and $\mathcal{M}_\lambda$, where $\mathcal{M}_\beta$ conducts out-of-subtext IDRR, while $\mathcal{M}_\lambda$ additionally uses subtext for in-subtext IDRR.

Assume $\mathcal{M}_\beta$ and $\mathcal{M}_\lambda$ output the relations R_β and R_λ respectively. The diagnoser first verifies the reliability of R_λ. Confidence score $\mathcal{C}$ is measured for verification. $\mathcal{C}$ is an average logistic probability over all the tokens output by $\mathcal{M}_\beta$. Each $c_i \in \mathcal{C}$ is the non-normalized probability estimated by the logistic function in the final layer of LLaMA3:

$$\mathcal{C} = \overline{\sum_{t_i \in R_\beta} log_{\mathcal{M}_\beta}(t_i)} \quad (1)$$

On this basis, the diagnoser verifies whether $\mathcal{C}$ is larger than the type-specific threshold θ. If larger, the diagnoser determines that R_λ is reliable for output, otherwise the prediction R_β of $\mathcal{M}_\beta$ is adopted. In our experiments, we provide an exclusive threshold for each relation type in the taxonomy of PDTB. They are obtained by empirical observation upon the IDRR performance obtained on the training set. More details of threshold settings and performance curves can be found in Appendix A.

3 Experiments

3.1 Dataset and Evaluation Metrics

We experiment on two versions of discourse relation analysis datasets, including PDTB-2.0 (Prasad et al., 2008) and 3.0 (Webber et al., 2019). We follow the previous work to use sections 0-22 for IDRR, where sections 2-20 are used for training, and sections 0-1 are used as the development set, while 21-22 for testing. Appendix B shows the data statistics in all the datasets.

Multi-class Macro-F1 (F_1) and accuracy rate (Acc) are used as the evaluation metrics.

3.2 Implementation Details

We use AdamW optimizer (Loshchilov and Hutter, 2019) to optimize LLaMA3. For subtext generation $\mathcal{M}_\alpha$, the learning rate is set to 1e-4. A 5-epoch training process is conducted. For the relation generators $\mathcal{M}_\beta$ and $\mathcal{M}_\lambda$, the learning rates are uniformly set to 5e-5, and the best checkpoint is reached based on F_1 within 10 epochs. Both

Method	Backbone Model	Parameters	PDTB2		PDTB3	
			F_1	Acc	F_1	Acc
ChatGPT (Chan et al., 2024)	GPT-3.5-turbo	-	36.11	44.18	-	-
PIDRA (Yung et al., 2024)	GPT-4	-	-	-	47.53	52.84
FCL (Long and Webber, 2022)	RoBERTa-base	125M	69.60	72.18	70.05	75.31
CP-KD (Wu et al., 2023)	RoBERTa-base	125M	68.86	75.43	72.07	77.00
CP-KD (Wu et al., 2023)	RoBERTa-large	355M	**71.88**	**76.77**	**75.52**	**78.56**
SCIDER (Cai et al., 2024)	RoBERTa-base	125M	67.00	72.11	-	-
OTMT (Jiang et al., 2024)	XLNet-large	355M	64.46	72.34	-	-
CG-T5 (Jiang et al., 2021)	T5-base	223M	57.18	65.54	-	-
DiscoPrompt (Chan et al., 2023)	T5-base	223M	65.79	71.70	-	-
DiscoPrompt (Chan et al., 2023)	T5-large	738M	70.84	75.65	-	-
IICOT (Lu et al., 2023)	Flan-T5-base	248M	65.26	71.13	69.79	73.98
IICOT (Lu et al., 2023)	Flan-T5-large	783M	69.23	76.04	73.06	**77.46**
Baseline	Llama3-8B-Instruct	8.03B	66.72	73.90	70.71	75.31
SCDN (ours)	Llama3-8B-Instruct	8.03B	**71.14**	**78.20**	**73.33**	76.93

Table 1: Performance on PDTB 2.0/3.0. Encoder-only PLMs, decoder-only LLMs and T5-based encoder-decoder models are considered. The best results are separately marked in bold for encoder-only and decoder-only models.

Model	PDTB2		PDTB3	
	F_1	Acc	F_1	Acc
Out-of-subtext	66.72	73.90	70.71	75.31
In-subtext	70.56	77.82	72.79	76.32
SCDN	**71.14**	**78.20**	**73.33**	**76.93**

Table 2: Test results in ablation study.

employ a weight decay of 1e-2, a batch size of 1, and gradient accumulation over 8 steps. We don't extensively tune the hyperparameters.

All experiments are performed on a NVIDIA A100 GPU. Our model implementations are based on PyTorch [2] and the Transformers library[3].

3.3 Main Results

We compare SCDN to the recently-proposed advanced models, including 1) decoder-only ChatGPT (Chan et al., 2024) and PIDRA (Yung et al., 2024), 2) encoder-only FCL (Long and Webber, 2022), CP-KD (Wu et al., 2023), SCIDER (Cai et al., 2024), and OTMT (Jiang et al., 2024), as well as 3) T5-based CG-T5 (Jiang et al., 2021), DiscoPrompt (Chan et al., 2023), and IICOT (Lu et al., 2023). The decoder-only models take full advantage of prompt engineering for IDRR. The encoder-only models are effective in representation learning for relation understanding. T5-based models combine the advantages. More contributions of these arts are summarized in Appendix C.

Table 1 shows the comparison results on the test set, where the 4-way relation classification performance on the main relation taxonomy is reported. It can be observed that our SCDN achieves higher $F1$-score than both the decoder-only and T5-based IDRR models. However, it still fails to outperform the encoder-only models. This is partially attributed to hallucination of LLaMA3 and off-topic results it generated.

Besides, we conduct experiments for the 2nd-level taxonomy, where each main relation type is divided into fine-grained relation senses. For example, the relation type "*Contingency*" contains the senses of "*Conditionality*" and "*Causality*". In this experiment, SCDN shows promising performance, which is reported in Appendix D due to page limit.

3.4 Ablation Study

To provide a direct insight into the influence of subtexts, we conduct an ablation study. Three IDRR models are considered in it, including 1) **out-of-subtext** generator $\mathcal{M}_\beta$ which is separately fine-tuned without using subtexts, 2) **in-subtext** generator $\mathcal{M}_\lambda$ which additioanlly uses the generated subtexts during fine-tuning, and 3) **SCDN** which uses both $\mathcal{M}_\beta$ and $\mathcal{M}_\lambda$, and reconciles them with the confidence-based diagnoser.

Table 2 shows the test results for the main relation taxonomy. It proves that the utilization of

[2] https://github.com/pytorch/pytorch
[3] https://github.com/huggingface/transformers

Subtext Generation Model	F_1	Acc
GPT-3.5-turbo	71.55	75.98
LLaMA3 w/o Distill	71.07	75.37
LLaMA3 w/ Distill (Partial)	71.66	76.19
LLaMA3 w/ Distill (Whole)	**72.79**	**76.32**

Table 3: Contributions from different subtext generators.

subtexts yields various levels of improvements. Appendix E provides a case study to show the effect.

3.5 Comparison among Subtext Generators

The qualified subtexts are crucial for SCDN. We investigate the subtexts generated by different LLMs, and verify their effects by our in-subtext model. There are three types of LLMs considered, including 1) GPT-3.5-turbo, 2) LLaMA3-8B-Instruct, 3) LLaMA3 which is strengthened by teacher-student knowledge distillation. During distillation, the subtexts generated by GPT-3.5-turbo for all training data are specified as **Guidance Data** from teacher. We use two different-sized guidance data: **Partial** and **Whole**. In the "Partial" case, we only adopt the guidance data which enables the subtext-based relation generator $\mathcal{M}_\lambda$ to output correct results. In the "Whole" case, all the guidance data is used.

Table 3 shows the performance of in-subtext IDRR models on the test set of PDTB 3.0, where different subtext generators are used. It can be observed that, compared to LLaMA3 (w/o distillation), GPT-3.5 enables the in-subtext model to perform better. Furthermore, no matter whether "Partial" or "Whole" guidance data is used, knowledge distillation causes improvements, and the latter case improves the in-subtext model more substantially. It is surprising that distillation allows the weaker LLaMA3 to be more contributive than its teacher GPT-3.5. The possible reason is because that LLaMA3 takes the advantage of itself when absorbing beneficial experience from GPT-3.5.

3.6 Prompts for Subtext Generation

The reliability of subtexts relies heavily on the design methods of prompts. For example, if we didn't remind LLMs of the ultimate purpose (i.e., being applied for IDRR), they fail to provide reliable subtexts. In our experiments, we evaluate different prompts as follows:

- P_1: It contains Q_1 and Q_2 (Section 2.1) that ask for subtext generation and IDRR in turn.

Prompt	LLM	F_1	Acc
P_1	GPT-3.5-turbo	35.88	41.79
P_2	GPT-3.5-turbo	38.30	42.76
P_3	GPT-3.5-turbo	**39.24**	**43.72**
P_1	GPT-4-turbo	44.84	50.07
P_2	GPT-4-turbo	**46.29**	**52.21**

Table 4: Reliability of Prompts (on PDTB 3.0).

- P_2: It replaces the key words in P_1 with their different synonyms, e.g., "*subtext*" is replaced with "*implicit meaning*".

- P_3: It expands P_2 by adding a prefix to Q_1, where the prefix is an additional question about "*whether there truly is a subtext in the considered argument*". This prompt helps to avoid a forcible subtext generation.

Table 4 shows the IDRR performance on the development set when the above prompts are separately used, where GPT-3.5 and 4.0 are considered during the validation process. It can be observed that both synonym replacement and non-forcible subtext generation yield improvements. Accordingly, P_2 has been introduced into our SCDN for optimization. However, P_3 fails to be used in SCDN as it actually causes performance degradation. For example, by P_3, the in-subtext "Single" generator $\mathcal{M}_\beta$ obtains a $F1$-score of 71.7% on the test set of PDTB 3.0, causing a performance reduction of about 1.1% (compared to the "Single" case in Table 2). This implies that LLaMA3 in $\mathcal{M}_\beta$ isn't able to effectively perform reasoning with a relatively-complex Chain-of-Thought (CoT). Besides, P_3 also causes severe performance reduction when it is used in SCDN. This is because that a limited number of subtexts are generated by GPT-3.5 due to the constraint from P_3, and thus GPT-to-LLaMA distillation falls into the low-resource scenario.

4 Conclusion

In this paper, we verify that the utilization of subtexts helps to strengthen the LLMs-based generative IDRR. Experiments demonstrate that reconciliation of in-subtext and out-of-subtext IDRR is effective. We also exhibit that distilling a lightweight LLM-based subtext generator is contributive, when the prompt doesn't raise a complex CoT.

In the future, we will investigate the default subtext which isn't implied in the given argument. On

the contrary, it derives from common-sense knowledge. Accordingly, we will convert binary arguments analysis to triplet, where the default subtext is regarded as the third non-negligible argument.

5 Limitations

This study proves the effectiveness of utilizing subtexts for enhancing IDRR. Nevertheless, a deeper analysis upon subtexts is still required. Our findings reveal that, in some cases, the shareable subtext is implied in one argument but irrelevant to the other, where the irrelevant argument appears as the noise during detecting the subtext. In some other cases, the default subtext occurs, which is not implied in any argument but derives from the common-sense knowledge. However, the subtext generation method in this paper cannot deal with these two problems. In the future, we will firstly study the common sense based default subtext generation. On this basis, we will convert the conventional binary arguments analysis to triplet, where the default subtext is used as a supplementary argument. This work will encounter the issues of 1) how to determine whether a pair of arguments are relevant to some default subtexts, and what they are, 2) how to detect and generate default subtexts conditioned on common-sense knowledge, and 3) how to reconcile the utilization of a triple of arguments and assign proper attention to them during relation discrimination.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under No.62376182.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Mingyang Cai, Zhen Yang, and Ping Jian. 2024. Improving implicit discourse relation recognition with semantics confrontation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8828–8839, Torino, Italia. ELRA and ICCL.

Chunkit Chan, Cheng Jiayang, Weiqi Wang, Yuxin Jiang, Tianqing Fang, Xin Liu, and Yangqiu Song. 2024. Exploring the potential of ChatGPT on sentence level relations: A focus on temporal, causal, and discourse relations. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 684–721, St. Julian's, Malta. Association for Computational Linguistics.

Chunkit Chan, Xin Liu, Jiayang Cheng, Zihan Li, Yangqiu Song, Ginny Wong, and Simon See. 2023. DiscoPrompt: Path prediction prompt tuning for implicit discourse relation recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 35–57, Toronto, Canada. Association for Computational Linguistics.

Zujun Dou, Yu Hong, Yu Sun, and Guodong Zhou. 2021. CVAE-based re-anchoring for implicit discourse relation classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1275–1283, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, and Angela Fan, et al. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Chengming Hu, Xuan Li, Dan Liu, Haolun Wu, Xi Chen, Ju Wang, and Xue Liu. 2023. Teacher-student architecture for knowledge distillation: A survey. *Preprint*, arXiv:2308.04268.

Yangfeng Ji and Jacob Eisenstein. 2015. One vector is not enough: Entity-augmented distributed semantics for discourse relations. *Transactions of the Association for Computational Linguistics*, 3:329–344.

Congcong Jiang, Tieyun Qian, and Bing Liu. 2024. One general teacher for multi-data multi-task: A new knowledge distillation framework for discourse relation analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:239–249.

Feng Jiang, Yaxin Fan, Xiaomin Chu, Peifeng Li, and Qiaoming Zhu. 2021. Not just classification: Recognizing implicit discourse relation on joint modeling of classification and generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2418–2431, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuxin Jiang, Linhan Zhang, and Wei Wang. 2023. Global and local hierarchy-aware contrastive framework for implicit discourse relation recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8048–8064, Toronto, Canada. Association for Computational Linguistics.

Wei Liu and Michael Strube. 2023. Annotation-inspired implicit discourse relation classification with auxiliary discourse connective generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15696–15712, Toronto, Canada. Association for Computational Linguistics.

Wei Liu, Stephen Wan, and Michael Strube. 2024. What causes the failure of explicit to implicit discourse relation recognition? *Preprint*, arXiv:2404.00999.

Wanqiu Long and Bonnie Webber. 2022. Facilitating contrastive learning of discourse relational senses by exploiting the hierarchy of sense relations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10704–10716, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Yuxiang Lu, Yu Hong, Zhipang Wang, and Guodong Zhou. 2023. Enhancing reasoning capabilities by instruction learning and chain-of-thoughts for implicit discourse relation recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5634–5640, Singapore. Association for Computational Linguistics.

Kazumasa Omura, Fei Cheng, and Sadao Kurohashi. 2024. An empirical study of synthetic data generation for implicit discourse relation recognition. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1073–1085, Torino, Italia. ELRA and ICCL.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Chenxu Wang, Ping Jian, and Mu Huang. 2023. Prompt-based logical semantics enhancement for implicit discourse relation recognition. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 687–699, Singapore. Association for Computational Linguistics.

Bonnie Webber, Rashmi Prasad, Alan Lee, and Aravind Joshi. 2019. The Penn Discourse TreeBank 3.0 annotation manual. *Philadelphia, University of Pennsylvania*, 35:108.

Changxing Wu, Liuwen Cao, Yubin Ge, Yang Liu, Min Zhang, and Jinsong Su. 2022. A label dependence-aware sequence generation model for multi-level implicit discourse relation recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11486–11494.

Hongyi Wu, Hao Zhou, Man Lan, Yuanbin Wu, and Yadong Zhang. 2023. Connective prediction for implicit discourse relation recognition via knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5908–5923, Toronto, Canada. Association for Computational Linguistics.

Wei Xiang, Bang Wang, Lu Dai, and Yijun Mo. 2022a. Encoding and fusing semantic connection and linguistic evidence for implicit discourse relation recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3247–3257, Dublin, Ireland. Association for Computational Linguistics.

Wei Xiang, Zhenglin Wang, Lu Dai, and Bang Wang. 2022b. ConnPrompt: Connective-cloze prompt learning for implicit discourse relation recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 902–911, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jing Xu, Ruifang He, Haodong Zhao, Huijie Wang, and Lei Zeng. 2023. Dual hierarchical contrastive learning for multi-level implicit discourse relation recognition. In *Natural Language Processing and Chinese Computing*, volume 14303, pages 55–66. Springer Nature Switzerland, Cham.

Frances Yung, Mansoor Ahmad, Merel Scholman, and Vera Demberg. 2024. Prompting implicit discourse relation annotation. *Preprint*, arXiv:2402.04918.

Lei Zeng, Ruifang He, Haowen Sun, Jing Xu, Chang Liu, and Bo Wang. 2024. Global and local hierarchical prompt tuning framework for multi-level implicit discourse relation recognition. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7760–7773, Torino, Italia. ELRA and ICCL.

Haodong Zhao, Ruifang He, Mengnan Xiao, and Jing Xu. 2023. Infusing hierarchical guidance into prompt tuning: A parameter-efficient framework for multi-level implicit discourse relation recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6477–6492, Toronto, Canada. Association for Computational Linguistics.

Hao Zhou, Man Lan, Yuanbin Wu, Yuefeng Chen, and Meirong Ma. 2022. Prompt-based connective prediction method for fine-grained implicit discourse relation recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3848–3858, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Relation	PDTB2-Top	PDTB3-Top
Comparison	31.06	29.97
Contingency	35.21	33.45
Expansion	31.22	29.83
Temporal	31.11	28.53

Table 5: Optimal thresholds for the main relation types.

A Threshold for Confidence Diagnosis

We set up a threshold for each relation type in the taxonomies of IDRR. For example, there are 4 thresholds provided for the four main IDRR relation types (i.e., *Expansion*, *Temporality*, *Contingency*, and *Comparison*). The adoption of different thresholds is because that the varying token lengths of relation labels cause unbalanced ranges of average confidence scores.

Let us consider the relation type T as an example. To seek for the optimal threshold θ_T for T, we empirically observe the T-oriented IDRR performance curve obtained when different optional values are used as thresholds. Within this empirical observation, the IDRR generator $\mathcal{M}_\beta$ and $\mathcal{M}_\lambda$ is used to predict relations and confidence scores for all the instances that hold a relation of T in the training set. And the accuracy Acc_T is used as the performance metric, which is calculated as follows:

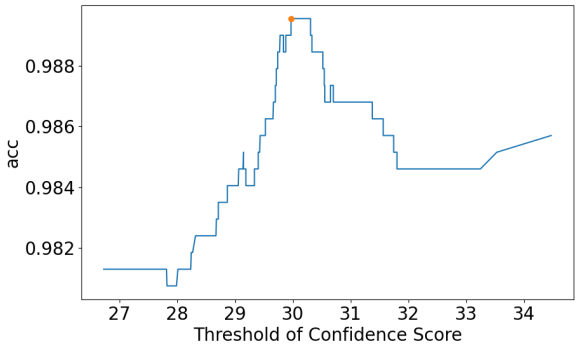
$$Acc_T(\check{\theta}) = \frac{n}{|D_T|} \tag{2}$$

where, $\check{\theta}$ is an optional threshold which is sequentially sampled from the range of confidence scores in the training set. n is the number of argument pairs which are given a positive relation prediction by $\mathcal{M}_\beta$ and $\mathcal{M}_\lambda$, and $|D_T|$ is the number of all argument pairs which hold the relation T.

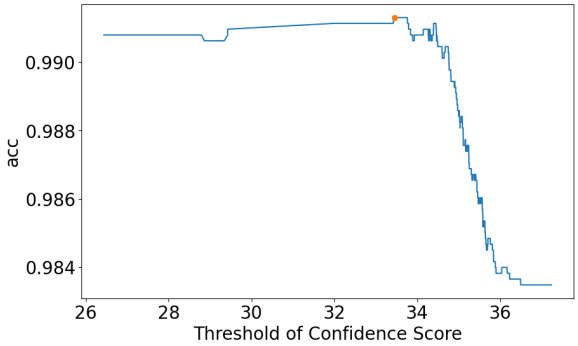
We adopt the optimal threshold θ by maximum likelihood estimation on all optional thresholds:

$$\theta_T = \arg\max_{\check{\theta}_T \in \check{\theta}_{all}} (Acc_T(\check{\theta}_T)) \tag{3}$$

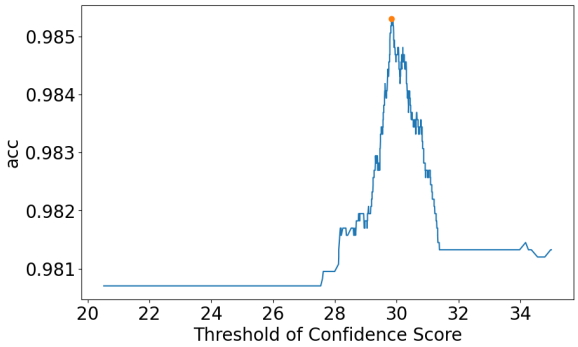
Figure 2 shows the curves of Acc_T changing with different thresholds in PDTB 3.0. We also present the specific values of the finally adopted thresholds θ in Table 5, 6 and 7, where Table 5 provides the thresholds for the main relation types (labeled as "Top"), while Table 6 and 7 give the thresholds for the relation senses in the secondary-level taxonomies (labeled as "Second").



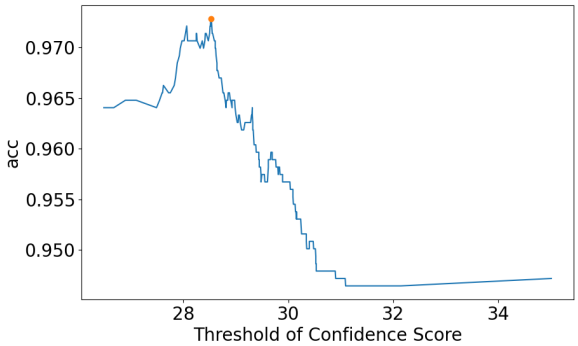
(a) Comparison



(b) Contingency



(c) Expansion



(d) Temporal

Figure 2: Acc_T on the training dataset with varying thresholds. The final selected thresholds are marked with red dots, which corresponds to the highest Acc_T.

Relation	PDTB2-Sec
Comparison.Concession	28.10
Comparison.Contrast	29.10
Contingency.Cause	30.79
Contingency.Pragmatic cause	29.05
Expansion.Alternative	31.68
Expansion.Conjunction	28.02
Expansion.Instantiation	30.80
Expansion.List	28.48
Expansion.Restatement	27.56
Temporal.Asynchronous	28.12
Temporal.Synchrony	28.25

Table 6: Thresholds for all relation senses of PDTB 2.0.

Relation	PDTB3-Sec
Comparison.Concession	28.01
Comparison.Contrast	30.75
Contingency.Cause	32.01
Contingency.Cause+Belief	29.41
Contingency.Condition	30.89
Contingency.Purpose	30.31
Expansion.Conjunction	27.16
Expansion.Equivalence	28.31
Expansion.Instantiation	30.05
Expansion.Level-of-detail	28.44
Expansion.Manner	29.08
Expansion.Substitution	27.96
Temporal.Asynchronous	28.03
Temporal.Synchronous	29.65

Table 7: Thresholds for all relation senses of PDTB 3.0.

B Statistics of PDTB datasets

We use the benchmark datasets PDTB-2.0 and 3.0 in our experiments, and follow the common practice (Ji and Eisenstein, 2015) to divide each of them into training (Train), validation (Dev) and test sets. The statistics in the datasets are shown in Table 8 and Table 9.

Relation	Train	Dev	Test
Comparison	1,894	191	146
Contingency	3,281	287	276
Expansion	6,792	651	556
Temporal	665	54	68
Total	12,632	1,183	1,046

Table 8: Data statistics of PDTB 2.0.

Relation	Train	Dev	Test
Comparison	1,830	190	154
Contingency	5,896	579	529
Expansion	7,941	748	643
Temporal	1,418	136	148
Total	17,085	1,653	1,474

Table 9: Data statistics of PDTB 3.0.

C Related Work

Recent research has demonstrated that PLMs outperform traditional machine learning methods for the IDRR task. Consequently, many studies have explored incorporating novel modules into the encoder-only transformer architecture to obtain better representations and extract more comprehensive features from the input. The new added modules includes Conditional Variational AutoEncoder (Dou et al., 2021), Graph Convolutional Network (Wu et al., 2022), Gated Recurrent Unit (Wu et al., 2022), and attention mechanism (Wu et al., 2022; Xiang et al., 2022a; Jiang et al., 2023).

On the other hand, some studies applied new training strategies like contractive learning (Long and Webber, 2022; Jiang et al., 2023; Xu et al., 2023; Zeng et al., 2024), knowledge distillation (Wu et al., 2023; Jiang et al., 2024), and extra pre-training (Wang et al., 2023).

Notably, Zhou et al. (2022) proposed a prompt-based approach that involves connective prediction and answer mapping. Their work paved the way for better leveraging of connectives, like enhancing the input with predicted connectives (Liu and Strube, 2023; Liu et al., 2024), or mapping the answers by connectives directly (Xiang et al., 2022b; Zhou et al., 2022; Wu et al., 2023; Wang et al., 2023; Zeng et al., 2024). Additionally, several studies investigated the potential of multi-level hierarchical information for IDRR. These works explored modeling the relationships between labels and fusing the global and local information within the multi-level hierarchical structure(Jiang et al., 2023; Xu et al., 2023; Zhao et al., 2023).

However, the potential of generative models and LLMs has been relatively underexplored for IDRR(Chan et al., 2024; Omura et al., 2024; Yung et al., 2024). Therefore, our work aims to address this gap by investigating how to effectively utilize LLMs' reasoning capabilities and incorporate additional relevant information into the input.

Method	Backbone Model	PDTB2-Sec		PDTB3-Sec	
		F_1	Acc	F_1	Acc
ChatGPT (Chan et al., 2024)	GPT-3.5-turbo	9.27	15.59	-	-
PIDRA (Yung et al., 2024)	GPT4	-	-	25.77	36.98
FCL (Long and Webber, 2022)	RoBERTa-base	49.66	61.69	57.62	64.68
CP-KD (Wu et al., 2023)	RoBERTa-base	44.77	64.00	50.12	66.21
CP-KD (Wu et al., 2023)	RoBERTa-large	47.78	66.41	52.16	67.84
SCIDER (Cai et al., 2024)	RoBERTa-base	-	59.62	-	-
OTMT (Jiang et al., 2024)	XLNet-large	-	61.06	-	-
CG-T5 (Jiang et al., 2021)	T5-base	37.76	-	-	-
DiscoPrompt (Chan et al., 2023)	T5-base	43.68	61.02	-	-
DiscoPrompt (Chan et al., 2023)	T5-large	49.03	64.58	-	-
SCDN (ours)	LLaMA3-8B-Instruct	46.38	62.46	55.04	64.35

Table 10: Performance of the secondary level classification on PDTB 2.0/3.0.

D Performance on Relation Senses

Besides of the relation types in the main-level taxonomy of PDTB, we additionally evaluate our models upon the secondary-level taxonomy. The latter taxonomy consists of the fine-grained relation senses. Table 10 shows the Macro $F1$-scores and accuracies of our models, as well as the previous work that reported the performance on the secondary-level taxonomy.

It can be found that our SCDN achieves promising performance, outperforming the T5-base based generative models. Nevertheless, SCDN has an obvious performance gap compared to the T5-large based DiscoPrompt (Chan et al., 2023). DiscoPrompt is a strong IDRR model which learns from the reliance between relations and implicit connectives during training. The implicit connectives are informative when being used as guidance during training, which allows T5-large to infer relations from additional perspectives. By contrast, we didn't use implicit connectives as guidance when fine-tuning SCDN. Besides, as shown in Table 11 and 12, some relation senses in the secondary-level taxonomy are given much less available training data than others in PDTB 2.0 and 3.0. More seriously, some arguments of such relation senses fail to be given a subtext by our subtext generator. This results in the insufficient training towards these relation senses when we fine-tune our in-subtext IDRR model and SCDN. And this causes severe performance degradation.

Relation	Number
Comparison.Concession	180
Comparison.Contrast	1,566
Contingency.Cause	3,227
Contingency.Pragmatic cause	51
Expansion.Alternative	146
Expansion.Conjunction	2,805
Expansion.Instantiation	1,061
Expansion.List	330
Expansion.Restatement	2,376
Temporal.Asynchronous	517
Temporal.Synchrony	147
Total	12,406

Table 11: Counts of secondary level relation types on the training set of PDTB 2.0.

E Case Study

To qualitatively assess the impact of subtexts on the reasoning ability of LLaMA3 for IDRR, we manually examine the test data as well as predictions of the **Out-of-subtext** and **In-subtext** models. The out-of-subtext IDRR model $\mathcal{M}_\beta$ predicts relations conditioned only on arguments, while the in-subtext IDRR model $\mathcal{M}_\lambda$ combines subtext and arguments, and uses them as clues for relation reasoning. The following analysis showcases the benefits associated with the use of subtexts:

Example 1

Argument 1: *I personally don't enjoy seeing players who I remember vividly from their playing days running about and being gallant about their deficiencies.*

Relation	Number
Comparison.Concession	1,170
Comparison.Contrast	639
Contingency.Cause	4,483
Contingency.Cause+Belief	157
Contingency.Condition	154
Contingency.Purpose	1,102
Expansion.Conjunction	3,584
Expansion.Equivalence	252
Expansion.Instantiation	1,117
Expansion.Level-of-detail	2,493
Expansion.Manner	191
Expansion.Substitution	278
Temporal.Asynchronous	985
Temporal.Synchronous	433
Total	17,038

Table 12: Counts of secondary level relation types on the training set of PDTB 3.0.

Argument 2: *People should be allowed to remember players as they were.*
Subtext: *The implicit meaning between the arguments is that people should be allowed to remember and cherish the memories of players from their playing days without seeing them participate in activities that may diminish their legacy or reputation. It suggests a desire to preserve the image and impact of these players in the eyes of the public.*
Ground-truth: Expansion
Ouf-of-subtext model: Comparison
In-subtext model: Expansion

Example 2

Argument 1: *For a family with taxable income of $50,000 to pay the same 28% incremental tax rate as a family with taxable income of $250,000.*
Argument 2: *The 33% bracket should apply to all income over the applicable level, not just the 5% rate adjustment amount.*
Subtext: *The implicit meaning between these arguments is that both are advocating for a more progressive tax system where higher-income individuals pay a higher percentage of their income in taxes. Argument 1 suggests that the current tax system is unfair because a family with a much lower income is paying the same percentage of taxes as a family with a much higher income. Argument 2 suggests that the tax brackets should be adjusted to ensure that higher-income individuals pay a higher tax rate on all of their income, not just the amount above a certain threshold. Both arguments imply that the current tax system is not equitable and needs to be restructured to address income inequality.*
Ground-truth: Comparison
Ouf-of-subtext model: Expansion
In-subtext model: Comparison

Example 3

Argument 1: *He believes in what he plays, and he plays superbly.*
Argument 2: *His recent appearance at the Metropolitan Museum, dubbed "A Musical Odyssey," was a case in point.*
Subtext: *The implicit meaning between these arguments is that the musician's belief in his music is evident in his exceptional performance at the Metropolitan Museum, showcasing his talent and skill as a musician.*
Ground-truth: Expansion.Instantiation
Ouf-of-subtext model: Expansion.Restatement
In-subtext model: Expansion.Instantiation

State-offset Tuning: State-based Parameter-Efficient Fine-Tuning for State Space Models

Wonjun Kang[1,2*] **Kevin Galim**[2*] **Yuchen Zeng**[3*] **Minjae Lee**[2]
Hyung Il Koo[2,4†] **Nam Ik Cho**[1]
[1] Seoul National University [2] FuriosaAI [3] UW-Madison [4] Ajou University
{kangwj1995, kevin.galim, minjae.lee, hikoo}@furiosa.ai, yzeng58@wisc.edu, nicho@snu.ac.kr

Abstract

State Space Models (SSMs) have emerged as efficient alternatives to Transformers, mitigating their quadratic computational cost. However, the application of Parameter-Efficient Fine-Tuning (PEFT) methods to SSMs remains largely unexplored. In particular, prompt-based methods like Prompt Tuning and Prefix-Tuning, which are widely used in Transformers, do not perform well on SSMs. To address this, we propose *state-based methods* as a superior alternative to prompt-based methods. This new family of methods naturally stems from the architectural characteristics of SSMs. State-based methods adjust state-related features directly instead of depending on external prompts. Furthermore, we introduce a novel state-based PEFT method: *State-offset Tuning*. At every timestep, our method directly affects the state at the current step, leading to more effective adaptation. Through extensive experiments across diverse datasets, we demonstrate the effectiveness of our method. Code is available at `https://github.com/furiosa-ai/ssm-state-tuning`.

1 Introduction

Large Language Models (LLMs) have gained significant attention for their strong performance in NLP tasks (Achiam et al., 2023; Brown et al., 2020), but suffer from the quadratic complexity of Transformer architectures (Vaswani et al., 2017). To mitigate this, subquadratic alternatives have gained interest (Katharopoulos et al., 2020; Peng et al., 2023; Sun et al., 2023), with State Space Models (SSMs) emerging as a promising solution (Gu and Dao, 2024; Dao and Gu, 2024).

Meanwhile, as LLMs scale up, full fine-tuning for downstream tasks becomes prohibitively expensive. Consequently, Parameter-Efficient Fine-Tuning (PEFT) (Houlsby et al., 2019; Hu et al., 2021; He et al., 2021; Zaken et al., 2022; Liu et al., 2021, 2022; Zeng and Lee, 2024) has emerged, which aims to reduce the number of trainable parameters while achieving adaptation performance comparable to full fine-tuning.

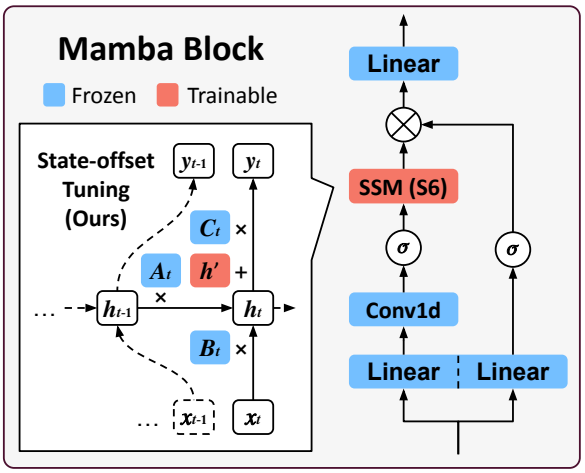


Figure 1: Illustration of our proposed State-offset Tuning on a Mamba block (Gu and Dao, 2024). State-offset Tuning injects a trainable state-offset $\boldsymbol{h}'$ at each timestep in the SSM module while keeping other parameters frozen, enabling parameter-efficient fine-tuning and improved downstream performance.

However, research on PEFT methods for SSMs remains limited despite their growing popularity. For instance, prompt-based PEFT methods, such as Prompt Tuning (Lester et al., 2021) and Prefix-Tuning (Li and Liang, 2021), have been widely applied to Transformers but fail to adapt effectively to SSMs (Galim et al., 2024). Therefore, new PEFT strategies tailored to SSMs are needed to fully leverage their architectural properties.

To bridge this gap, we introduce *state-based PEFT methods* that leverage the intrinsic properties of SSMs, offering a superior alternative to prompt-based methods. Building on this concept, we propose *State-offset Tuning*. This method directly ad-

*Equal contribution.
†Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 474–487
July 27 - August 1, 2025

justs the state-related features rather than relying on external prompts, enabling more effective adaptation.

In summary, our main contributions are:

- We introduce *state-based methods*, a new family of PEFT techniques for SSMs, offering a superior alternative to prompt-based approaches.
- We propose *State-offset Tuning* as a new state-based PEFT method.
- We demonstrate the effectiveness of our method through experiments on a variety of datasets, consistently outperforming existing fine-tuning techniques.

2 Related Works

2.1 State Space Models

Linear State-Space Layers (LSSL) are one of the earliest applications of SSMs in sequence modeling (Gu et al., 2021), leveraging HiPPO (Gu et al., 2020) to initialize the state matrix. However, its high computational overhead limits practicality. Gu et al. (2022) introduced Structured State Space Models (S4), which mitigate this by structuring the state matrix. Recently, Mamba (Gu and Dao, 2024; Dao and Gu, 2024) enhanced modeling capabilities by introducing an input-dependent S6 block.

2.2 Parameter-Efficient Fine-Tuning

In this section, we review existing PEFT methods. For more details, see Sec. D.

Parameter-based Methods One approach to parameter-based PEFT methods is to selectively fine-tune specific layers within the model while keeping the remaining layers frozen. BitFit (Zaken et al., 2022) is a lightweight and effective strategy that focuses solely on fine-tuning a model's bias terms. Furthermore, LoRA (Hu et al., 2021) represents a notable parameter-based PEFT method by introducing low-rank matrices for weight updates, facilitating efficient adaptation.

Prompt-based Methods Instead of fine-tuning model parameters, Prompt Tuning (Lester et al., 2021) enhances models by prepending trainable soft embeddings to the prompt. Prefix-Tuning (Li and Liang, 2021) builds on this approach by injecting trainable embeddings into each Transformer layer, achieving strong adaptation results for Transformer-based LLMs.

PEFT for SSMs Concurrently, Galim et al. (2024) showed that LoRA outperforms prompt-based methods on SSMs. Furthermore, they proposed Selective Dimension Tuning (SDT) for fine-tuning the SSM module while applying LoRA on the linear projection matrices when fine-tuning Mamba models. Yoshimura et al. (2025) suggested a new PEFT method called Additional-scan, which increases the hidden state dimension of SSMs, fine-tuning only its additional parameters.

3 PEFT Methods on SSMs

SSM Preliminaries Assuming a single channel dimension, SSMs such as S4 (Gu et al., 2022) transform a signal $x_t \in \mathbb{R}$ into $y_t \in \mathbb{R}$ through an H-dimensional latent state $\boldsymbol{h}_t \in \mathbb{R}^H$ as below:

$$\boldsymbol{h}_t = \overline{\boldsymbol{A}}\boldsymbol{h}_{t-1} + \overline{\boldsymbol{B}}x_t, \qquad y_t = \boldsymbol{C}\boldsymbol{h}_t,$$

where $\overline{\boldsymbol{B}} \in \mathbb{R}^{H\times 1}$ controls input influence, $\overline{\boldsymbol{A}} \in \mathbb{R}^{H\times H}$ governs state dynamics, and $\boldsymbol{C} \in \mathbb{R}^{1\times H}$ maps the state to the output. $\overline{\boldsymbol{A}}$ and $\overline{\boldsymbol{B}}$ represent discretized versions of $\boldsymbol{A}$ and $\boldsymbol{B}$, parameterized by a learnable step size $\Delta \in \mathbb{R}$.

In S6 (the SSM module of Mamba), input dependency is integrated by using input-dependent $\overline{\boldsymbol{A}}_t$, $\overline{\boldsymbol{B}}_t$, and $\boldsymbol{C}_t$ at every timestep. Specifically, given D channels with $\boldsymbol{x}_t \in \mathbb{R}^D$, learnable parameters $\boldsymbol{W_B}, \boldsymbol{W_C} \in \mathbb{R}^{H\times D}$, and $\boldsymbol{W_\Delta} \in \mathbb{R}^{D\times D}$ compute $\boldsymbol{B}_t = \boldsymbol{W_B}\boldsymbol{x}_t$, $\boldsymbol{C}_t = \boldsymbol{W_C}\boldsymbol{x}_t$, and $\boldsymbol{\Delta} = \boldsymbol{W_\Delta}\boldsymbol{x}_t$. In this section, we consider S4 for simplicity.

3.1 Prompt-based PEFT Methods on SSMs

Prefix-Tuning Can Update Only the Initial State of an SSM Generally, SSMs assume that the initial hidden state is $\boldsymbol{h}_0 = \boldsymbol{0}$. We can express $\boldsymbol{h}_t$ with $\boldsymbol{h}_0$ as $\boldsymbol{h}_t = \sum_{i=1}^{t} \overline{\boldsymbol{A}}^{t-i}\overline{\boldsymbol{B}}_i x_i + \overline{\boldsymbol{A}}^t \boldsymbol{h}_0$.

Assume we have virtual tokens $x_{(-V+1)}, \ldots, x_0$. If we prepend virtual tokens as prefix to the input sequence, we can write the updated $\widehat{\boldsymbol{h}}_t$ as below:

$$\widehat{\boldsymbol{h}}_t = \boldsymbol{h}_t + \overline{\boldsymbol{A}}^t \sum_{i=0}^{V-1} \overline{\boldsymbol{A}}^i \overline{\boldsymbol{B}} x_{-i} = \boldsymbol{h}_t + \overline{\boldsymbol{A}}^t \boldsymbol{h}_{\text{prefix}}.$$

By introducing a non-zero $\widehat{\boldsymbol{h}}_0$, we can substitute $\widehat{\boldsymbol{h}}_0$ for $\boldsymbol{h}_{\text{prefix}}$ making Prefix-Tuning, or optimizing virtual tokens, equivalent to updating the initial state. As optimized virtual tokens only affect the initial state $\widehat{\boldsymbol{h}}_0$, Prefix-Tuning's expressivity is upper-bounded by updating the initial state directly (Galim et al., 2024). Since Prefix-Tuning is an extended version of Prompt Tuning, this upper bound is applicable to Prompt Tuning as well.

Galim et al. (2024) showed Initial State Tuning, an advanced version of Prefix-Tuning, which directly optimizes the channel-specific initial state $\boldsymbol{h}' \in \mathbb{R}^H$, resulting in DH trainable parameters in total across all D channels. The updated output $\widehat{y}_t$ for Initial State Tuning can be written as in Table 1.

3.2 State-based Methods: A New Family of PEFT Methods for SSMs

We define state-based methods as a new family of PEFT methods specifically designed for SSMs. These methods directly modify the intrinsic state-related features within the SSM module.

In contrast, prompt-based methods, such as Prefix-Tuning, influence the hidden state of the SSM module indirectly by introducing external virtual tokens. While both approaches adjust the hidden state of the SSM module, state-based methods operate within the SSM module itself, offering a more direct and expressive adaptation strategy.

Based on our definition, we classify Initial State Tuning as a state-based method. While Initial State Tuning surpasses Prefix-Tuning (Galim et al., 2024), it still falls short compared to other fine-tuning methods on SSMs. To bridge this gap, we propose a novel state-based method for enhanced performance.

Initial State Tuning	$\widehat{y}_t = y_t + \boldsymbol{C}_t(\prod_{i=1}^{t} \overline{\boldsymbol{A}}_i)\boldsymbol{h}'$
State-offset Tuning ($\boldsymbol{h}$)	$\widehat{y}_t = y_t + \boldsymbol{C}_t\boldsymbol{h}'$
State-offset Tuning (y)	$\widehat{y}_t = y_t + y'$

Table 1: State-based methods for S6. Our methods eliminate the time-dependent coefficient $\prod_{i=1}^{t} \overline{\boldsymbol{A}}_i$, ensuring a uniform effect across timesteps.

Prompt-based	Timestep T	Timestep $T+1$
Prefix	$[\textbf{prefix}, x_1, \ldots, x_T]$	$\rightarrow [\textbf{prefix}, x_1, \ldots, x_T, \boldsymbol{x}_{T+1}]$
Suffix	$[x_1, \ldots, x_T, \textbf{suffix}]$	$\rightarrow [x_1, \ldots, x_T, \textbf{suffix}, x_{T+1}]$
Iterative Suffix	$[x_1, \ldots, x_T, \textbf{suffix}]$	$\rightarrow [x_1, \ldots, x_T, x_{T+1}, \textbf{suffix}]$

Table 2: Comparison of Prefix-Tuning, Suffix-Tuning, and Iterative Suffix-Tuning.

4 Proposed State-based PEFT Method

In this section, we propose *State-offset Tuning* as a new state-based PEFT method. A visual comparison with Initial State Tuning and Prefix-Tuning is provided in Sec. A.

4.1 State-offset Tuning

Initial State Tuning introduces an additional term $\boldsymbol{h}'$ with a coefficient $\overline{\boldsymbol{A}}^t$ for S4 and $\prod_{i=1}^{t} \overline{\boldsymbol{A}}_i$ for S6. However, this coefficient, which varies for each timestep, tends to decrease over time, leading to inconsistent effects. This is related to the issue that SSMs struggle to recall early tokens (Fu et al., 2022). To address this and ensure a consistent effect for each timestep, we introduce *State-offset Tuning*, which eliminates this coefficient.

State-offset Tuning adds a constant, learnable state-offset $\boldsymbol{h}'$ to the hidden state $\boldsymbol{h}$ before obtaining the updated output $\widehat{y}_t$ (Fig. 1). Therefore, unlike Initial State Tuning, State-offset Tuning does not alter the hidden state dynamics directly. Instead, State-offset Tuning adds a constant $\boldsymbol{h}'$ repetitively for each timestep, ensuring a uniform impact.

We formulate State-offset Tuning ($\boldsymbol{h}$) for S6 in Table 1, where we optimize $\boldsymbol{h}' \in \mathbb{R}^H$. In S4, $\boldsymbol{C}_t$ does not depend on the input, simplifying to a constant $\boldsymbol{C}$. This allows us to optimize a bias y' instead of $\boldsymbol{h}'$ (with $y' := \boldsymbol{C}\boldsymbol{h}'$ for each dimension). We name this method State-offset Tuning (y). For S4, State-offset Tuning (y) and State-offset Tuning ($\boldsymbol{h}$) are equivalent. In S6, opting for the simpler State-offset Tuning (y) enhances parameter efficiency by decreasing the tunable parameters from DH to D.

4.2 Connection to Prompt-based Methods

To further validate the methodology of State-offset Tuning, we examine its connection to prompt-based methods and demonstrate its correspondence to *Iterative Suffix-Tuning*.

Iterative Suffix-Tuning Li and Liang (2021) showed that in Transformers, inserting virtual tokens at the beginning (Prefix-Tuning) or the end (Suffix-Tuning, referred to as Infix-Tuning in their work) yields similar performance.

However, for SSMs, the position of the inserted virtual tokens is crucial, as these models tend to forget early tokens. The effect of Prefix-Tuning and Suffix-Tuning diminishes as the model processes subsequent timesteps. This leads to the question: how can we maintain consistent influence of virtual tokens across all timesteps in SSMs?

To achieve this, we propose Iterative Suffix-Tuning. As shown in Table 2, both Prefix-Tuning and Suffix-Tuning hold virtual tokens in fixed positions throughout all timesteps. Conversely, Iterative Suffix-Tuning shifts virtual tokens to the sequence's last position at each timestep, ensur-

Model Size		Mamba 1.4B									Mamba 130M			
Dataset		Params	Spider					SAMSum			Params	DART		GLUE
Type	Method	(%)	All	Easy	Medium	Hard	Extra	R1	R2	RL	(%)	MET.	BLEU	Avg.
-	Pretrained	0.00	0.0	0.0	0.0	0.0	0.0	10.9	1.5	10.2	0.00	18.1	1.2	41.0
	Full Fine-tuning (All)	100.00	66.2	84.3	69.5	53.4	43.4	51.2	27.3	42.9	100.00	71.0	51.8	80.5
	Full Fine-tuning (S6)	4.46	56.7	76.6	57.8	46.0	34.9	51.1	26.9	42.2	4.31	70.3	48.7	79.3
Parameter based	LoRA	0.46	56.3	75.0	56.5	**50.6**	**33.7**	50.5	26.4	42.2	0.92	69.9	**50.8**	78.3
	BitFit	0.03	51.3	74.2	50.9	43.1	26.5	50.3	25.7	41.9	0.06	67.0	43.7	77.9
	Additional-scan	0.34	26.9	44.4	25.6	21.3	10.2	37.6	17.5	30.9	0.68	60.6	15.8	62.4
Prompt based	Prompt Tuning	0.01	43.6	65.3	42.4	33.3	25.3	50.1	25.6	41.6	0.04	66.2	39.8	63.8
	Prefix-Tuning	12.81	39.7	65.7	38.6	31.0	15.1	50.6	**26.5**	42.1	22.69	66.6	42.5	68.6
State based	Initial State Tuning	0.23	51.8	**77.8**	51.1	35.1	32.5	50.0	26.0	41.3	0.45	69.1	46.2	77.4
	State-offset Tuning ($\boldsymbol{h}$)	0.23	**57.4**	77.4	**59.9**	44.8	**33.7**	**50.9**	**26.5**	**42.4**	0.45	**70.0**	47.0	**78.5**
	State-offset Tuning (y)	0.01	53.0	77.4	55.4	40.8	22.9	50.6	26.1	42.0	0.03	66.8	45.2	77.7

Table 3: Experimental results for fine-tuning the SSM module (S6) of Mamba (Gu and Dao, 2024) models. We assess Spider and its subsets using execution accuracy, SAMSum with ROUGE-1/2/L scores, DART using METEOR and BLEU scores, and GLUE by calculating the average score. To demonstrate the effectiveness of our methods, we configure the hyperparameters of each method to ensure their parameter budget is comparable to or exceeds that of our methods. **Bold** and underline indicate the best and the second-best results, respectively, among all methods (excluding full fine-tuning). Our State-offset Tuning ($\boldsymbol{h}$) outperforms all other methods on most datasets, and our State-offset Tuning (y) shows comparable or better performance than other methods despite its significantly fewer trainable parameters.

ing uniform influence in SSMs. This method is akin to how State-offset Tuning eliminates the time-varying coefficient in Initial State Tuning, enforcing a consistent effect at every timestep. We show that Iterative Suffix-Tuning in SSMs is equivalent to State-offset Tuning (as detailed in Sec. B).

5 Experiments

5.1 Experiment Setup

We conduct experiments for fine-tuning the SSM module (S6) using pretrained Mamba (Gu and Dao, 2024) and Mamba-2 (Dao and Gu, 2024) models on four datasets: Spider (Yu et al., 2018), SAMSum (Gliwa et al., 2019), DART (Nan et al., 2021), and GLUE (Wang et al., 2019). For further information on datasets, evaluation metrics, and experimental details, refer to Secs. E and F. We use LoRA (Hu et al., 2021), BitFit (Zaken et al., 2022), and Additional-scan (Yoshimura et al., 2025) as parameter-based methods. For prompt-based methods, we employ Prompt Tuning (Lester et al., 2021) and Prefix-Tuning[1] (Li and Liang, 2021). For state-based methods, we utilize Initial State Tuning (Galim et al., 2024), along with our proposed methods, State-offset Tuning ($\boldsymbol{h}$) and State-offset Tuning (y).

[1]Following Yoshimura et al. (2025), our Prefix-Tuning implementation for SSMs corresponds to their Affix-Tuning formulation, adapted specifically for SSM architectures.

5.2 Experimental Results

Table 3 shows the results on Mamba models. Additional results, including Mamba-2 results, are provided in Sec. G. In the appendix, we further compare the training speed, training memory usage, and computational overhead during inference between LoRA and State-offset Tuning ($\boldsymbol{h}$). Our findings show that State-offset Tuning ($\boldsymbol{h}$) is faster, more memory-efficient, and introduces lower FLOP overhead compared to LoRA. Additionally, we evaluate the performance of State-offset Tuning ($\boldsymbol{h}$) within SSMs against Prefix-Tuning in Transformers, further highlighting the effectiveness of our approach.

State-based Methods Outperform Prompt-based Methods Table 3 shows that all state-based methods outperform prompt-based methods, supporting the claim that state-based methods are superior to prompt-based methods on SSMs.

In particular, our State-offset Tuning ($\boldsymbol{h}$) achieves the best results among all tested PEFT methods on most datasets. Our State-offset Tuning (y) outperforms Initial State Tuning on most datasets, using just 0.01% of the parameters compared to 0.23% by Initial State Tuning.

State-offset Tuning Outperforms Parameter-Based Methods State-offset Tuning ($\boldsymbol{h}$) outperforms BitFit across all datasets and surpasses LoRA on most datasets. Notably, it also outperforms Additional-scan, a method specifically designed for fine-tuning SSM modules, across all datasets.

Furthermore, State-offset Tuning ($\boldsymbol{h}$) achieves performance comparable to full fine-tuning (S6), highlighting the effectiveness of state-based PEFT for SSM modules, despite using significantly fewer parameters. The results from Mamba-2 (Table 11) further validate the effectiveness of our method. We also include a comparison to Selective Dimension Tuning (SDT) (Galim et al., 2024) in Sec. G.4, showing that our method outperforms SDT while using fewer parameters.

6 Conclusion

In this paper, we introduce *state-based methods* as a new family of PEFT methods for State Space Models, serving as a superior alternative to prompt-based methods. We propose *State-offset Tuning* as a new state-based PEFT method and demonstrate its effectiveness through extensive experiments.

7 Limitations

While we demonstrate that State-offset Tuning is effective for fine-tuning SSMs in the text domain, its applicability to other domains, such as vision or speech, remains unexplored. Existing PEFT methods, such as LoRA and Prompt Tuning, have been successfully applied across various domains (Jia et al., 2022; Gal et al., 2023; Ran et al., 2024). Extending State-offset Tuning to models in other domains, such as Vision Mamba (Zhu et al., 2025), is an interesting direction for future work.

Potential Risks Our approach enables parameter-efficient fine-tuning (PEFT) of pretrained SSMs, significantly reducing the computational cost of adaptation. While this is beneficial for resource-constrained scenarios, it also presents potential risks. Specifically, adversaries could leverage our method to efficiently fine-tune pretrained SSMs on harmful or biased data, enabling the rapid adaptation of models for malicious purposes with minimal computational resources. This could lead to the proliferation of harmful or deceptive models that reinforce misinformation, bias, or toxicity. To mitigate these risks, future work should explore more robust safety measures, such as integrating ethical fine-tuning constraints and monitoring mechanisms.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. arXiv preprint arXiv:2303.08774.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In Proceedings of the second PASCAL challenges workshop on recognising textual entailment, volume 1. Citeseer.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. TAC, 7(8):1.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In International Conference on Machine Learning, pages 2397–2430. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In Advances in Neural Information Processing Systems, volume 33, pages 1877–1901.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In Machine learning challenges workshop, pages 177–190. Springer.

Tri Dao and Albert Gu. 2024. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In International Conference on Machine Learning.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In Third international workshop on paraphrasing (IWP2005).

Daniel Y Fu, Tri Dao, Khaled Kamal Saab, Armin W Thomas, Atri Rudra, and Christopher Re. 2022. Hungry hungry hippos: Towards language modeling with state space models. In International Conference on Learning Representations.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-Or. 2023. An image is worth one word: Personalizing text-to-image generation using textual inversion. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net.

Kevin Galim, Wonjun Kang, Yuchen Zeng, Hyung Il Koo, and Kangwook Lee. 2024. Parameter-efficient fine-tuning of state space models. arXiv preprint arXiv:2410.09016.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB dataset of diverse text for language modeling. arXiv preprint arXiv:2101.00027.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing, pages 1–9.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. EMNLP-IJCNLP 2019, page 70.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. In First Conference on Language Modeling.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. 2020. Hippo: Recurrent memory with optimal polynomial projections. In Advances in Neural Information Processing Systems, volume 33, pages 1474–1487.

Albert Gu, Karan Goel, and Christopher Re. 2022. Efficiently modeling long sequences with structured state spaces. In International Conference on Learning Representations.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. 2021. Combining recurrent, convolutional, and continuous-time models with linear state space layers. In Advances in Neural Information Processing Systems, volume 34, pages 572–585.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. In International Conference on Learning Representations.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In International Conference on Machine Learning, pages 2790–2799.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. LoRA: Low-rank adaptation of large language models. In International Conference on Learning Representations.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. 2022. Visual prompt tuning. In European Conference on Computer Vision, pages 709–727. Springer.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are RNNs: Fast autoregressive transformers with linear attention. In International Conference on Machine Learning, pages 5156–5165.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3045–3059.

Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4582–4597.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 61–68.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT Understands, Too. arXiv:2103.10385.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, et al. 2021. DART: Open-Domain Structured Data Record to Text Generation. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 432–447.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. pages 311–318.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. Preprint, arXiv:1912.01703.

Bo Peng, Eric Alcaide, Quentin Gregory Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Nguyen Chung, Leon Derczynski, et al. 2023. RWKV: Reinventing RNNs for the transformer era. In The 2023 Conference on Empirical Methods in Natural Language Processing.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Lingmin Ran, Xiaodong Cun, Jia-Wei Liu, Rui Zhao, Song Zijie, Xintao Wang, Jussi Keppo, and Mike Zheng Shou. 2024. X-adapter: Adding universal compatibility of plugins for upgraded diffusion model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8775–8784.

Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Lianmin Zheng, Kurt Keutzer, et al. 2023. S-lora: Serving thousands of concurrent lora adapters. CoRR.

Daniel G. A. Smith and Johnnie Gray. 2018. opt_einsum - a python package for optimizing contraction order for einsum-like expressions. Journal of Open Source Software, 3(26):753.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing, pages 1631–1642.

Vladislav Sovrasov. 2018-2024. ptflops: a flops counting tool for neural networks in pytorch framework.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive network: A successor to transformer for large language models. arXiv preprint arXiv:2307.08621.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In International Conference on Learning Representations.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. Transactions of the Association for Computational Linguistics, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018, pages 1112–1122. Association for Computational Linguistics (ACL).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Masakazu Yoshimura, Teruaki Hayashi, and Yota Maeda. 2025. MambaPEFT: Exploring parameter-efficient fine-tuning for mamba. In The Thirteenth International Conference on Learning Representations.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3911–3921.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1–9.

Yuchen Zeng and Kangwook Lee. 2024. The expressive power of low-rank adaptation. In International Conference on Learning Representations.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. 2025. Vision mamba: Efficient visual representation learning with bidirectional state space model. In Forty-first International Conference on Machine Learning.

A Visual Comparison of Prompt-based Methods and State-based Methods

Fig. 2 compares prompt-based methods and state-based methods, including our proposed State-offset Tuning, within the S6 block.

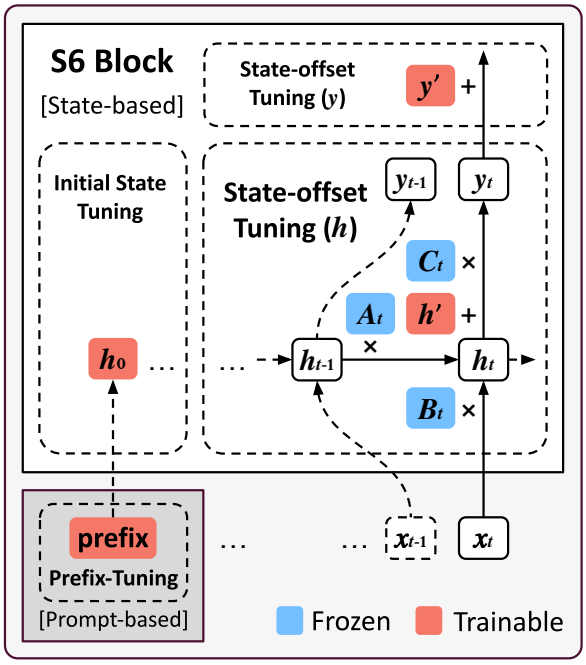


Figure 2: Visual comparison of prompt-based methods and state-based methods in the S6 block.

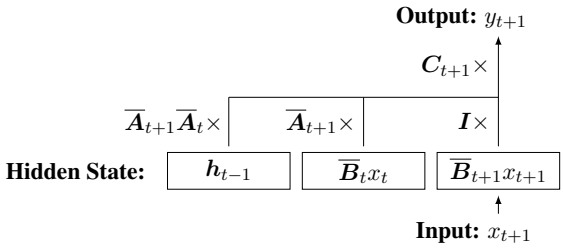
State-based Methods Operate within the SSM Module Fig. 2 shows that prompt-based methods, such as Prefix-Tuning, rely on virtual tokens external to the S6 block. In contrast, state-based methods, such as Initial State Tuning, State-offset Tuning ($\boldsymbol{h}$), and State-offset Tuning (y), directly adjust state-related features within the S6 block.

State-offset Tuning Affects the Current Timestep Figure 2 illustrates how Prefix-Tuning and Initial State Tuning modify features at early timesteps, indirectly affecting the current state. However, this impact diminishes over time. In contrast, State-offset Tuning ($\boldsymbol{h}$) and State-offset Tuning (y) directly influence the state at each timestep, resulting in more effective adaptation.

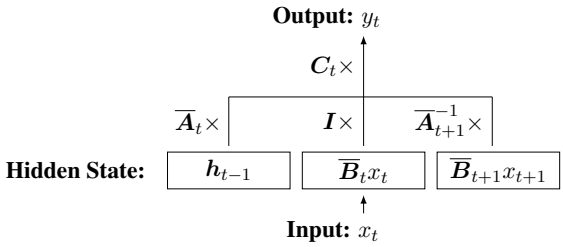
B Iterative Suffix-Tuning and State-offset Tuning

In this section, we show that Iterative Suffix-Tuning for SSMs is equivalent to State-offset Tuning.

State-offset Tuning is Iterative Suffix-Tuning Fig. 3 provides two different implementations of



(a) Iterative Suffix-Tuning (with $t+1$ as current timestep)



(b) Iterative Suffix-Tuning (with t as current timestep)

Figure 3: Two different implementations of Iterative Suffix-Tuning in S6. We show that Fig. 3b is equivalent to State-offset Tuning.

Iterative Suffix-Tuning on SSMs (S6) with virtual token (suffix) x_{t+1}. Fig. 3a views $t+1$ as current timestep. In this case, input-dependent $\boldsymbol{C}_{t+1} = \boldsymbol{W}_{\boldsymbol{C}}\boldsymbol{x}_{t+1}$ is determined solely by the suffix $\boldsymbol{x}_{t+1} \in \mathbb{R}^D$, which is constant at inference time, thus the input dependency of $\boldsymbol{C}$ is lost, reducing the expressive power of S6.

To address this, we view t as current timestep instead and interpret x_{t+1} as future token (Fig. 3b). Consequently, we time-shift x_{t+1} by multiplying it with the inverse of $\overline{\boldsymbol{A}}_{t+1}$.

$$\text{Fig. 3a:} \quad y_{t+1} = \boldsymbol{C}_{t+1}(\overline{\boldsymbol{A}}_{t+1}\boldsymbol{h}_t + \overline{\boldsymbol{B}}_{t+1}x_{t+1}),$$

$$\text{Fig. 3b:} \quad y_t = \boldsymbol{C}_t(\boldsymbol{h}_t + \overline{\boldsymbol{A}}_{t+1}^{-1}\overline{\boldsymbol{B}}_{t+1}x_{t+1}).$$

Therefore, according to the equation corresponding to Fig. 3b, Iterative Suffix-Tuning can be implemented by updating only $\overline{\boldsymbol{A}}_{t+1}^{-1}\overline{\boldsymbol{B}}_{t+1}x_{t+1}$. Since this term depends solely on the constant suffix x_{t+1}, we can directly replace it with a learnable parameter $\boldsymbol{h}'$ ($\boldsymbol{h}' := \overline{\boldsymbol{A}}_{t+1}^{-1}\overline{\boldsymbol{B}}_{t+1}x_{t+1}$), which is equivelant to State-offset Tuning ($\boldsymbol{h}$) (Table 1).

C Low-Rank State-offset Tuning

State-offset Tuning ($\boldsymbol{h}$) shows superior parameter efficiency on Mamba versus other PEFT methods. To further reduce trainable parameters, we can represent the learnable state-offset as a product of two low-rank matrices, inspired by LoRA (Hu et al., 2021). This is particularly useful for Mamba-2,

where the state dimension is larger than in Mamba, leading to an increased number of trainable parameters. In such cases, low-rank techniques can effectively mitigate the parameter overhead. Experimental results of State-offset Tuning ($\boldsymbol{h}$) with lower rank on Mamba-2 are provided in Sec. G.2.

D PEFT Baselines

In this section, we provide a more detailed description of the baseline methods.

LoRA (Hu et al., 2021) LoRA aims to fine-tune large models by maintaining the bulk of pretrained parameters untouched while introducing trainable low-rank matrices within each Transformer's layer. This method leverages linear algebra principles where a large matrix can be effectively approximated by two low-rank matrices, thus reducing the number of parameters. LoRA includes a scaling parameter to adjust the influence of original and LoRA weights during training. We use the Hugging Face version (Apache License 2.0, Mangrulkar et al. (2022)) of LoRA for our experiments.

Prompt Tuning (Lester et al., 2021) This method involves freezing the entire model and adding a trainable soft prompt to the input. The prompt consists of continuous virtual tokens that provide additional context.

Prefix-Tuning (Li and Liang, 2021) Similar to Prompt Tuning, Prefix-Tuning adds trainable tokens but extends them across every Transformer layer by appending trainable embeddings to the attention matrices. To combat the instability in training these prefixes, an over-parameterized MLP is utilized, which can be discarded after training.

BitFit (Zaken et al., 2022) This PEFT method simplifies fine-tuning by training only the bias terms while freezing the other model weights, drastically reducing trainable parameters.

SDT (Galim et al., 2024) SDT (Selective Dimension Tuning) employs a sparse updating approach for the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ ($\boldsymbol{W_B}$ and $\boldsymbol{W_C}$ for S6), while additionally applying LoRA to the linear projection layers. All remaining layers are kept frozen. The process for determining which parameters to update involves a warmup stage, during which parameters are flagged as updatable if they exhibit a significant gradient magnitude. In our SDT experiments, we excluded LoRA from the linear projection layers and focused solely on its S6 component.

Additional-scan (Yoshimura et al., 2025) This approach enhances the model's expressivity by expanding the state dimensions for $\boldsymbol{A}$, $\boldsymbol{W_B}$, and $\boldsymbol{W_C}$. During training, only the added dimensions are marked as trainable.

E Datasets

Dataset	#Train	#Valid	#Epochs	Model size	Metrics
RTE	2490	277	10	130m	Acc.
MRPC	3668	408	10	130m	Acc.
CoLA	8551	1043	10	130m	Acc.
SST-2	67349	872	10	130m	Acc.
QNLI	104743	5463	10	130m	Acc.
QQP	363846	40430	3	130m	Acc.
MNLI	392702	19647	3	130m	Acc.
Spider	6918	1034	10	1.4B, 2.8B	Acc.
SAMSum	14732	819	10	1.4B	ROUGE
DART	62659	2768	10	130m	METEOR, BLEU

Table 4: Dataset details. We report the number of training and validation samples, number of training epochs, employed model size and evaluation metrics.

This paper examines four datasets across two domains: Natural Language Understanding (NLU) and Natural Language Generation (NLG). Table 4 presents detailed information for each dataset.

GLUE (Wang et al., 2019) A benchmark comprising nine tasks in English for assessing language understanding models, including sentiment analysis, linguistic acceptability, and question answering. We use the the following datasets: RTE (Dagan et al., 2005; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009), MRPC (Dolan and Brockett, 2005), CoLA (Warstadt et al., 2019), SST-2 (Socher et al., 2013), QNLI (Rajpurkar et al., 2018), QQP[2], and MNLI (Williams et al., 2018). Evaluation is mainly through accuracy, except for CoLA where Matthews correlation is used. The final metric is calculated as the average accuracy (Matthews correlation for CoLA) across all datasets. The individual datasets are available under different permissive licenses. We use the version hosted at `https://huggingface.co/datasets/nyu-mll/glue`.

SAMSum (Gliwa et al., 2019) A dataset for dialogue summarization featuring about 16,000 synthetic conversations in English with summaries, created to simulate digital communications with

[2] `https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs`

varied tones and styles. Its structure helps in developing systems that process conversational text. The dataset is evaluated via ROUGE score. This dataset is available under the CC BY-NC-ND 4.0 license. We use the version hosted at https://huggingface.co/datasets/Samsung/samsum.

Spider (Yu et al., 2018) A text-to-SQL dataset with 10,000 annotated SQL queries across 200+ databases, classifying queries from easy to extra hard based on SQL operation complexity. It involves translating English questions to SQL, evaluated via execution accuracy. Execution accuracy considers the output correct if the model's predicted SQL query and the ground truth SQL query yield the same results when executed on the database. This dataset is available under the CC BY-SA 4.0 license. We use the version hosted at https://huggingface.co/datasets/xlangai/spider.

DART (Nan et al., 2021) Comprising over 80,000 instances, DART focuses on English RDF-to-text generation, organized by structured data triples and corresponding text summaries. It is assessed using METEOR and BLEU metrics. This dataset is available under the MIT license. We use the version hosted at https://huggingface.co/datasets/Yale-LILY/dart.

F Experimental Details

For every dataset, we select the model size based on how difficult the dataset is and conduct a brief grid search for one epoch using a subset of the data (1k-2k instances) with learning rates of $\{4 \times 10^{-1}, 2 \times 10^{-1}, 1 \times 10^{-1}, ..., 1 \times 10^{-5}\}$. The best learning rate is then selected as the rate that has the lowest training loss. In our experimental results, we report the metric from the best epoch observed on the validation set during training, employing early stopping. Each experiment is conducted once. We apply fine-tuning methods to the SSM module (S6) of Mamba (130M, 1.4B, 2.8B)[3] and the SSM module (SSD) of Mamba-2 (130M, 1.3B)[4] pretrained from Pile (MIT License, Gao et al. (2020)) using AdamW (Loshchilov and Hutter, 2019) with a linear decay schedule for the learning rate. In general, we choose hyperparameters for each individual method to ensure that all methods operate within a similar parameter budget. Tables 5 and 6 show selected learning rates and chosen hyperparameters for each method. For assessing NLG tasks, we utilize beam search with five beams and a maximum beam length of 1024. BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005) metrics are computed using Hugging Face's evaluate library[5].

We use an NVIDIA RTX 3090 24GB for training models with less than 1 billion parameters, and an NVIDIA H100 80GB for larger models. We implemented our project in PyTorch (Modified BSD license, Paszke et al. (2019)), utilizing the Hugging Face trainer (Apache License 2.0, Wolf et al. (2020)). We train with batch size 4 for 10 epochs on all datasets except QQP and MNLI for which we use 3 epochs, allowing each training run to finish in under 16 hours. This project spanned three months, utilizing four NVIDIA RTX 3090 24GB GPUs and four NVIDIA H100 80GB GPUs, totaling approximately 17,000 GPU hours.

G Additional Experimental Results

G.1 Mamba Results

Training Speed and Memory Usage We conduct a small experiment to compare the memory usage and training speed of State-offset Tuning ($\boldsymbol{h}$) and LoRA, as they performed most similarly in terms of dataset metrics in our experiments. Using a single H100 GPU, we train for 100 batch iterations with a batch size of 4 and a 1K context, continuously measuring memory usage and batch latency.

Table 7 shows the training speed and maximum memory usage for different Mamba sizes for State-offset Tuning ($\boldsymbol{h}$) and LoRA. State-offset Tuning ($\boldsymbol{h}$) uses less memory and is faster, even with more trainable parameters. In this experiment, we selected hyperparameters to ensure LoRA has less trainable parameters than State-offset Tuning ($\boldsymbol{h}$). We believe State-offset Tuning ($\boldsymbol{h}$)'s efficiency stems from our optimized einsum implementation, enhanced with the opt_einsum (MIT License, Smith and Gray (2018)) Python package to reduce memory usage and improve latency.

[3]Apache License 2.0, https://huggingface.co/state-spaces/mamba-{130m,1.4b,2.8b}

[4]Apache License 2.0, https://huggingface.co/state-spaces/mamba2-{130m,1.3b}

[5]Apache License 2.0, https://huggingface.co/spaces/evaluate-metric/{bleu,rouge,meteor}

Model	Mamba										Mamba-2		
Method / Dataset	RTE	MRPC	CoLA	SST-2	QNLI	QQP	MNLI	DART	SAMSum	Spider	DART	SAMSum	Spider
LoRA	2e-03	2e-03	4e-05	2e-03	1e-03	1e-03	2e-03	4e-03	2e-03	4e-03	4e-03	2e-03	4e-03
Additional-scan	4e-03	2e-03	2e-03	1e-01	2e-03	4e-02	4e-03	4e-03	4e-03	4e-03	2e-02	4e-03	1e-02
SDT	1e-03	4e-02	1e-01	4e-02	2e-02	2e-02	1e-01	4e-02	2e-02	4e-02	-	-	-
Initial State Tuning	4e-04	1e-03	2e-03	2e-03	2e-03	2e-03	2e-03	2e-03	2e-04	1e-03	4e-03	2e-04	4e-04
State-offset Tuning ($\boldsymbol{h}$)	1e-03	2e-04	2e-04	1e-04	1e-04	4e-05	4e-04	4e-04	1e-04	2e-04	1e-03	2e-05	2e-05
State-offset Tuning ($\boldsymbol{h}$) (low rank)	-	-	-	-	-	-	-	-	-	-	4e-03	2e-04	2e-04
State-offset Tuning (y)	1e-03	2e-03	1e-03	1e-03	2e-03	1e-03	1e-03	4e-03	1e-03	2e-03	1e-02	2e-04	1e-03

Table 5: Learning rates for each method and dataset. For Mamba and Mamba-2, learning rates for each method and dataset are determined via a small grid search on a dataset subset. The learning rate yielding the best training loss is chosen as the final rate.

Method /Model	Mamba 130M	Mamba 1.4B	Mamba-2 130M	Mamba-2 1.3B
LoRA	Rank = 8 α = 8 Dropout = 0.1 Modules = all weight matrices in S6	Rank = 8 α = 8 Dropout = 0.1 Modules = all weight matrices in S6	Rank = 16 α = 16 Dropout = 0.1 Modules = all weight matrices in SSD	Rank = 16 α = 16 Dropout = 0.1 Modules = all weight matrices in SSD
Additional-scan	#States = 8	#States = 8	#States = 32	#States = 32
SDT	Freeze #Channels = 50.0% Freeze #States = 75.0%	Freeze #Channels = 50.0% Freeze #States = 75.0%	-	-
Initial State Tuning	-	-	-	-
State-offset Tuning ($\boldsymbol{h}$)	-	-	-	-
State-offset Tuning ($\boldsymbol{h}$) (low rank)	-	-	Rank = 32	Rank = 64
State-offset Tuning (y)	-	-	-	-

Table 6: Hyperparameter settings for each model and PEFT method. In general, we adjust hyperparameters to maintain a similar number of trainable parameters.

Model	**Method**	**Params** (%)	**Mem.** (GB)	**Latency** (s)
130M	State-offset Tuning ($\boldsymbol{h}$)	0.45	**4.2**	**0.13**
	LoRA	0.35	5.44	0.18
370M	State-offset Tuning ($\boldsymbol{h}$)	0.42	**9.36**	**0.33**
	LoRA	0.32	11.56	0.45
790M	State-offset Tuning ($\boldsymbol{h}$)	0.3	**13.91**	**0.49**
	LoRA	0.23	17.17	0.61
1.4B	State-offset Tuning ($\boldsymbol{h}$)	0.23	**18.77**	**0.67**
	LoRA	0.17	22.99	0.8
2.8B	State-offset Tuning ($\boldsymbol{h}$)	0.19	**31.49**	**1.13**
	LoRA	0.14	37.84	1.33

Table 7: Training speed and memory usage. For each Mamba size, we compare the maximum memory usage and mean latency for processing a single batch during training. Our State-offset Tuning ($\boldsymbol{h}$) is compared against LoRA, as it demonstrated the most similar performance in the experiment section. We configure LoRA to use fewer trainable parameters than State-offset Tuning ($\boldsymbol{h}$). Despite this, State-offset Tuning ($\boldsymbol{h}$) still consumes less memory and is faster in training.

FLOP Overhead While it is possible to avoid extra FLOP with LoRA in constrained single-task settings by merging weights into the pretrained model, real-world serving scenarios often require a single pretrained model to support multiple downstream tasks simultaneously via multiple LoRA adapters. In such cases, avoiding extra FLOP would require storing separately merged models for each task in memory—an inefficient solution. Alternatively, merging weights dynamically at inference time introduces significant computational bottlenecks. As a result, many recent works focus on serving many LoRA adapters efficiently without weight merging (Sheng et al., 2023).

Sequence Length		**L=128**	**L=256**	**L=512**	**L=1024**	**Relative**
Model	**Method**	**GFLOP**				(%)
130M	Pretrained	16.45	32.90	65.81	131.61	100.000
	State-offset Tuning ($\boldsymbol{h}$)	16.46	32.91	65.83	131.65	+ 0.029
	LoRA	16.61	33.21	66.42	132.84	+ 0.937
370M	Pretrained	47.35	94.69	189.39	378.77	100.000
	State-offset Tuning ($\boldsymbol{h}$)	47.36	94.72	189.44	378.87	+ 0.027
	LoRA	47.76	95.52	191.03	382.06	+ 0.867
790M	Pretrained	101.22	202.44	404.88	809.75	100.000
	State-offset Tuning ($\boldsymbol{h}$)	101.24	202.48	404.95	809.90	+ 0.019
	LoRA	101.84	203.67	407.34	814.67	+ 0.608
1.4B	Pretrained	175.23	350.45	700.90	1401.79	100.000
	State-offset Tuning ($\boldsymbol{h}$)	175.25	350.50	701.00	1401.99	+ 0.014
	LoRA	176.05	352.09	704.17	1408.35	+ 0.468
2.8B	Pretrained	353.66	707.32	1414.63	2829.25	100.000
	State-offset Tuning ($\boldsymbol{h}$)	353.70	707.40	1414.80	2829.59	+ 0.012
	LoRA	355.03	710.05	1420.09	2840.17	+ 0.386

Table 8: FLOP overhead across various model sizes and sequence lengths. State-offset Tuning adds less than 0.03% overhead, whereas LoRA incurs over 30× more extra FLOP compared to ours.

Given these practical considerations, we evaluate LoRA without weight merging and conduct experiments comparing the additional FLOP of LoRA and our State-offset Tuning method during inference. We use ptflops (Sovrasov, 2018-2024) to measure computational overhead. As shown in Table 8, our method adds less than 0.03% overhead, while LoRA results in more than 30 times

the additional FLOP compared to ours. These results highlight the superior FLOP efficiency of our method compared to LoRA.

Mamba 2.8B Results Table 9 shows the experimental results using Mamba 2.8B. Our State-offset Tuning ($\boldsymbol{h}$) outperforms all methods except full fine-tuning.

Mamba Results on GLUE Dataset Table 10 shows the full results on the GLUE dataset using Mamba 130M. Our State-offset Tuning ($\boldsymbol{h}$) achieves the highest average score among all PEFT methods.

G.2 Mamba-2 Results

Table 11 shows experimental results with Mamba-2 (Dao and Gu, 2024) models. State-offset Tuning ($\boldsymbol{h}$) with low-rank adaptation (Sec. C) significantly reduces the number of trainable parameters. It outperforms existing methods on the Spider benchmark by a large margin and achieves performance comparable to other approaches on the SAMSum and DART datasets.

G.3 State-offset Tuning in SSMs vs. Prefix-Tuning in Transformers

To highlight the effectiveness of State-offset Tuning, we compare its performance with Prefix-Tuning on the Transformer model Pythia (Biderman et al., 2023). We conduct full fine-tuning and Prefix-Tuning experiments on Pythia 160M on GLUE tasks. The results are shown in Table 12.

Full fine-tuning on Mamba 130M generally surpasses Pythia 160M, consistent with Gu and Dao (2024). Prefix-Tuning on both Mamba and Pythia reaches about 85–90% of their full fine-tuning performance.

Our State-offset Tuning achieves approximately 98% of full fine-tuning performance, effectively closing the gap. This success highlights its precise design for SSM-based models.

G.4 Comparison to Selective Dimension Tuning (SDT)

We additionally compare our method with Selective Dimension Tuning (SDT) (Galim et al., 2024), a technique derived from theoretical analysis of SSMs. Note that the hyperparameter selection differs from that used in Galim et al. (2024) to ensure the parameter count is more comparable to ours. As shown in Table 13, our method consistently outperforms SDT in most cases while using fewer parameters.

Model Size		Mamba 2.8B					
Dataset		Params	Spider				
Type	Method	(%)	All	Easy	Medium	Hard	Extra
-	Full Fine-tuning (All)	100.00	71.8	87.5	73.5	63.8	51.8
	Full Fine-tuning (S6)	4.44	65.7	81.9	68.8	58.0	41.0
Parameter based	LoRA	0.38	63.9	86.3	**68.2**	49.4	34.3
	BitFit	0.02	59.9	82.3	60.8	**52.9**	31.3
	Additional-scan	0.28	35.0	62.0	31.9	27.4	12.1
Prompt based	Prompt Tuning	0.01	50.7	75.4	53.8	37.4	19.3
	Prefix-Tuning	10.82	45.1	75.0	45.1	32.2	13.9
State based	Initial State Tuning	0.19	59.7	82.3	62.3	43.7	35.5
	State-offset Tuning ($\boldsymbol{h}$)	0.19	**65.0**	**89.1**	65.9	51.7	**40.4**
	State-offset Tuning (y)	0.01	63.1	85.9	64.1	52.3	37.3

Table 9: Experimental results of fine-tuning the SSM module using pretrained Mamba 2.8B. State-offset Tuning ($\boldsymbol{h}$) stands out as the most effective method among all PEFT approaches.

Model Size		Mamba 130M								
Dataset		Params	GLUE							
Type	Method	(%)	RTE	MRPC	CoLA	SST-2	QNLI	QQP	MNLI	Avg.
-	Full Fine-tuning (All)	100.00	71.1	80.6	63.2	92.2	87.4	87.9	80.8	80.5
	Full Fine-tuning (S6)	4.31	69.7	78.9	59.1	91.5	88.1	87.5	80.5	79.3
Parameter based	LoRA	0.92	66.1	78.7	**57.8**	90.8	**87.8**	**86.9**	**79.8**	78.3
	BitFit	0.06	69.5	80.4	54.7	92.0	86.2	85.3	77.2	77.9
	Additional-scan	0.68	57.9	74.0	38.6	79.0	79.9	70.5	36.9	62.4
Prompt based	Prompt Tuning	0.04	56.0	71.6	12.0	89.4	76.8	79.6	61.5	63.8
	Prefix-Tuning	22.69	67.5	75.7	43.4	91.5	83.4	83.1	35.6	68.6
State based	Initial State Tuning	0.45	66.8	78.4	53.0	**92.4**	86.4	86.1	78.5	77.4
	State-offset Tuning ($\boldsymbol{h}$)	0.45	67.4	**80.8**	56.2	91.9	87.7	85.6	79.7	**78.5**
	State-offset Tuning (y)	0.03	**70.0**	79.6	52.5	91.7	86.3	85.6	78.2	77.7

Table 10: Full results of fine-tuning the SSM module on the GLUE dataset using pretrained Mamba 130M. Our State-offset Tuning ($\boldsymbol{h}$) achieves the highest average score among all PEFT methods.

Model Size		Mamba-2 1.3B									Mamba-2 130M		
Dataset		Params	Spider					SAMSum			Params	DART	
Type	Method	(%)	All	Easy	Medium	Hard	Extra	R1	R2	RL	(%)	MET.	BLEU
-	Full Fine-tuning (All)	100.00	64.8	85.9	65.7	54.0	42.2	51.0	26.9	42.5	100.00	66.6	34.9
	Full Fine-tuning (SSD)	2.42	55.1	76.2	56.1	42.5	34.3	50.5	26.3	42.4	4.17	65.7	39.7
Parameter based	LoRA	0.37	45.4	69.0	44.4	37.4	21.1	49.7	25.9	41.7	0.76	**70.3**	**49.6**
	BitFit	0.02	50.9	71.4	51.6	45.4	24.1	**50.9**	26.5	**42.6**	0.03	66.2	39.0
	Additional-scan	0.47	31.9	57.3	30.5	23.0	7.2	43.0	20.1	34.8	0.91	58.5	16.0
Prompt based	Prompt Tuning	0.01	45.2	62.5	46.9	34.5	25.9	49.6	26.1	41.6	0.04	65.5	36.9
	Prefix-Tuning	6.99	47.4	71.0	48.2	32.2	25.9	50.8	26.5	**42.6**	12.81	69.2	46.5
State based	Initial State Tuning	1.84	54.3	73.4	57.2	45.4	27.1	50.4	26.4	42.3	3.53	65.3	37.2
	State-offset Tuning ($\boldsymbol{h}$)	1.84	58.5	**79.3**	61.6	44.6	**33.7**	48.8	24.7	40.5	3.53	70.0	46.3
	State-offset Tuning ($\boldsymbol{h}$) (low rank)	0.35	**60.5**	79.0	**65.7**	**52.3**	27.7	50.4	**26.8**	42.5	0.72	69.8	47.9
	State-offset Tuning (y)	0.01	43.6	66.5	42.1	36.9	21.1	50.3	26.2	42.2	0.03	65.9	38.7

Table 11: Experimental results of fine-tuning the SSM module using pretrained Mamba-2 (Dao and Gu, 2024) models. We evaluate Spider and its subsets with execution accuracy, SAMSum using ROUGE-1/2/L scores, and DART through METEOR and BLEU scores. State-offset Tuning ($\boldsymbol{h}$) with low-rank adaptation (Sec. C) significantly reduces trainable parameters. It outperforms existing methods on Spider by a wide margin and matches the performance of other approaches on SAMSum and DART.

Dataset		Params	GLUE								
Model	**Method**	(%)	**RTE**	**MRPC**	**CoLA**	**SST-2**	**QNLI**	**QQP**	**MNLI**	**Avg.**	**Relative (%)**
Pythia 160M	Full Fine-tuning	100.00	64.3	77.0	20.5	88.7	85.0	88.8	79.2	71.9	100
	Prefix-Tuning	8.36	57.4	75.0	4.6	88.2	81.5	80.6	62.2	64.2	89
Mamba 130M	Full Fine-tuning	100.00	71.1	80.6	63.2	92.2	87.4	87.9	80.8	80.5	100
	Prefix-Tuning	22.69	67.5	75.7	43.4	91.5	83.4	83.1	35.6	68.6	85
	State-offset Tuning (h)	0.45	67.4	80.8	56.2	91.9	87.7	85.6	79.7	78.5	98

Table 12: Prefix-Tuning experiments on Pythia 160M and Mamba 130M on GLUE tasks. State-offset Tuning for Mamba achieves approximately 98% of full fine-tuning performance, while Prefix-Tuning reaches about 85–90% in both SSM and Transformer architectures.

Model Size	**Mamba 1.4B**									**Mamba 130M**			
Dataset	**Params**	**Spider**					**SAMSum**			**Params**	**DART**		**GLUE**
Method	(%)	All	Easy	Medium	Hard	Extra	R1	R2	RL	(%)	MET.	BLEU	Avg.
SDT	0.26	19.8	38.3	16.6	16.1	4.8	46.3	21.5	37.7	0.51	67.5	**48.2**	63.7
State-offset Tuning (h)	0.23	**57.4**	**77.4**	**59.9**	**44.8**	**33.7**	**50.9**	**26.5**	**42.4**	0.45	**70.0**	47.0	**78.5**
State-offset Tuning (y)	0.01	53.0	**77.4**	55.4	40.8	22.9	50.6	26.1	42.0	0.03	66.8	45.2	77.7

Table 13: Comparison with Selective Dimension Tuning (SDT) (Galim et al., 2024) on Spider, SAMSum, DART, and GLUE. Our method outperforms SDT in most cases while using fewer parameters. Note that the hyperparameter configuration of SDT differs from that in Galim et al. (2024) to ensure a more comparable parameter count.

Internal and External Impacts of Natural Language Processing Papers

Yu Zhang
Texas A&M University
yuzhang@tamu.edu

Abstract

We investigate the impacts of NLP research published in top-tier conferences (*i.e.,* ACL, EMNLP, and NAACL) from 1979 to 2024. By analyzing citations from research articles and external sources such as patents, media, and policy documents, we examine how different NLP topics are consumed both within the academic community and by the broader public. Our findings reveal that language modeling has the widest internal and external influence, while linguistic foundations have lower impacts. We also observe that internal and external impacts generally align, but topics like ethics, bias, and fairness show significant attention in policy documents with much fewer academic citations. Additionally, external domains exhibit distinct preferences, with patents focusing on practical NLP applications and media and policy documents engaging more with the societal implications of NLP models.

1 Introduction

"*Is ACL an AI conference?*" Comments from opinion leaders[1][2] within the natural language processing (NLP) community definitely matter. Meanwhile, a complementary perspective is how the broader academic community and the public perceive and utilize papers from top-tier NLP conferences (*e.g.,* ACL, EMNLP, and NAACL). We can categorize the public use of NLP papers into two types: (1) *Internal Use*: NLP papers are cited by other research articles in not only NLP but also various other fields (Zhang et al., 2024), due to the ideas, methods, datasets, or findings they present (Jurgens et al., 2018; Cohan et al., 2019); and (2) *External Use*: Research can break out of the academic "ivory tower" (Cao et al., 2023) and diffuse into the technological, societal, and governmental discussions (Yin et al., 2022). Specifically, NLP papers can be referenced by patents, media posts, policy documents, and text in other public channels. If we infer how public audiences define these NLP conferences based on which papers from the conferences are used by them, the question becomes: Papers of which topics (*e.g.,* cutting-edge AI models or linguistic foundations) are more extensively consumed internally and externally?

To answer this question, in this paper, we present a comprehensive scientometric study examining the internal and external impacts of NLP papers published in ACL, EMNLP, and NAACL between 1979 and 2024. We collect data from various sources (Priem et al., 2022; Marx and Fuegi, 2022; Adie and Roe, 2013; Szomszor and Adie, 2022) to obtain up-to-date references to NLP papers from internal (*i.e.,* research articles) and external (*i.e.,* patents, news/social media, and policy documents) domains. Based on the collected data, we calculate the average number of times that papers of a specific topic (*e.g.,* "*Language Modeling*" or "*Ethics, Bias, and Fairness*") are used in an internal/external domain, and employ it as an indicator to measure the impact of the topic within that domain.

Our analysis leads to the following key observations: First, among all research topics that NLP conferences are calling for, language modeling demonstrates the broadest internal and external impacts, whereas linguistic foundations, such as phonology, morphology, psycholinguistics, and pragmatics, exhibit relatively low influence. Second, the internal and external impacts of an NLP topic generally align, though there are outliers, such as "*Ethics, Bias, and Fairness*", which show notably high use in policy documents but are cited far less frequently in research articles. Third, different external domains favor different NLP topics, offering complementary insights into an NLP paper's internal impact. Patents tend to consume practical NLP applications, while media and policy documents are

[1] https://faculty.washington.edu/ebender/papers/ACL_2024_Presidential_Address.pdf
[2] https://gist.github.com/yoavg/f952b7a6cafd2024f44c8bc444a64315

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 488–494
July 27 - August 1, 2025

more likely attracted by the behavior and societal influence of NLP models.

2 Data

We first describe how we collect data from multiple sources to measure the internal impact of NLP papers and their external uses across various domains, including patents, media, and policy documents.

NLP Papers: ACL Anthology. From the ACL Anthology[3], we extract all papers whose venue is marked as ACL, EMNLP, or NAACL (including main conference papers, findings papers, system demonstrations, industry track papers, *etc.*). We choose these three conferences because CSRankings designates them as "top conferences" for ranking purposes, and they are the top three in Google Scholar's "Computational Linguistics" category. In total, we obtain 24,821 papers published between 1979 and 2024 in these three conferences.

Internal Impact (Citation): OpenAlex. Because Microsoft Academic (Sinha et al., 2015) terminated its service at the end of 2021, to get the up-to-date number of citations for each paper, we link our extracted NLP papers to OpenAlex (Priem et al., 2022)[4]. In all, we successfully map 21,104 papers to the OpenAlex database and get their numbers of citations. We use $\mathcal{P}$ to denote the set of these mapped papers. We should mention that each paper $p \in \mathcal{P}$ may have multiple versions (*e.g.,* a preprint version and an NLP conference version). When calculating the internal or external impact of p, we merge all versions into a single record by summing their citation counts.

External Impact (Patent): Reliance on Science. To study references to NLP papers in patents, we follow previous studies (Yin et al., 2022; Cao et al., 2023) and utilize the Reliance on Science dataset (Marx and Fuegi, 2020, 2022)[5], which contains references from USPTO patents to OpenAlex papers. In total, we identify 20,218 links from patents to papers in $\mathcal{P}$.

External Impact (Media): Altmetric. To quantify how NLP papers are mentioned in media, we rely on the data provided by Altmetric (Adie and Roe, 2013)[6], which archives references from media, including both news and social media (*i.e.,* Twitter, Facebook, Reddit, and blogs), to academic papers. In all, we obtain 18,586 media-to-paper links for the NLP publications in $\mathcal{P}$.

External Impact (Policy Document): Overton. To examine references to NLP papers in policy documents, we follow Yin et al. (2021) and exploit the Overton database (Szomszor and Adie, 2022)[7]. Overton defines "policy documents" as "research, briefs, reviews, or reports written with the goal of influencing or changing policy" from "governments, public bodies, IGOs, NGOs and think tanks". We obtain 1,223 links from policy documents to papers in $\mathcal{P}$.

3 Analysis

3.1 Internal and External Impacts by Topic

We adopt the 25 submission topics (excluding the special theme) outlined in the ACL 2025 Call for Papers[8] as our topic space for NLP papers, denoted as $\mathcal{T}$. For each NLP paper $p \in \mathcal{P}$, we use GPT-4o (Hurst et al., 2024) to predict its most relevant topic $t_p \in \mathcal{T}$. Human evaluation is then conducted to assess the quality of GPT-4o annotations. We find that GPT-4o gives reliable annotations, and the human evaluators share "substantial agreement" according to the Fleiss' kappa (Fleiss, 1971). One can refer to Appendix A for more details about the annotation process. Based on the annotations, we use $\mathcal{P}_t = \{p : p \in \mathcal{P} \text{ and } t_p = t\}$ to denote the set of NLP papers labeled with the topic t ($\forall t \in \mathcal{T}$).

To quantify the impact of NLP papers with a certain topic $t \in \mathcal{T}$ in a certain domain $d \in \{\texttt{Citation}, \texttt{Patent}, \texttt{Media}, \texttt{PolicyDocument}\}$, we consider the following metric:

$$\text{Impact}(t|d) = \frac{\sum_{p \in \mathcal{P}_t} \#\text{citation}(p|d)/|\mathcal{P}_t|}{\sum_{p \in \mathcal{P}} \#\text{citation}(p|d)/|\mathcal{P}|}.$$

Here, $\#\text{citation}(p|d)$ represents the "citation count" (in a broader sense) of paper p given domain d. To be specific, when $d = \texttt{Citation}$, it is the number of times that p is cited by other academic papers (from the entire OpenAlex database rather than ACL Anthology only); when $d \in \{\texttt{Patent}, \texttt{Media}, \texttt{PolicyDocument}\}$, it is the number of times that p is mentioned by patents, media posts, and policy documents, respectively, in our

[3] https://github.com/acl-org/acl-anthology, accessed on November 25, 2024.

[4] https://openalex.s3.amazonaws.com/RELEASE_NOTES.txt, the version released on November 25, 2024.

[5] https://zenodo.org/records/11461587, the version released on June 3, 2024.

[6] Please refer to https://www.altmetric.com to request access to the data. We use the version updated in August 2023.

[7] Please refer to https://www.overton.io to request access to the data. We extracted the data on December 8, 2024.

[8] https://2025.aclweb.org/calls/main_conference_papers

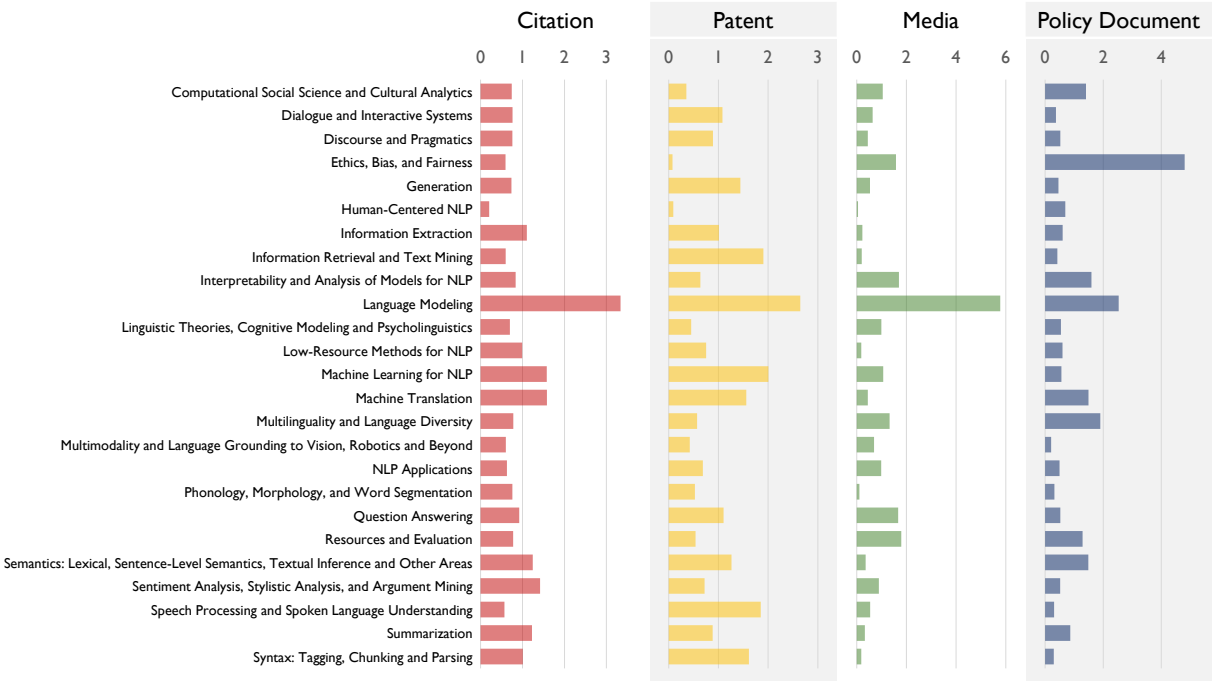


Figure 1: Impact of NLP papers with a certain topic in a certain domain.

collected data. $\text{Impact}(t|d)$ is the average "citation count" of papers with topic t given domain d, normalized by the average "citation count" of all papers given domain d. Our proposed metric is inspired by Yin et al. (2022), with one key difference: we consider the actual number of times a paper p is cited, whereas Yin et al. (2022) binarize this value, focusing only on whether p is cited or not (*i.e.,* $\mathbf{1}_{\#\text{citation}(p|d)>0}$).

Figure 1 visualizes $\text{Impact}(t|d)$ across all topics and domains considered by us, from which we have the following key observations:

Observation 1: Papers on language modeling have a broad impact across all internal and external domains.

The topic "*Language Modeling*" enjoys the highest impact in the `Citation`, `Patent`, and `Media` domains, as well as the second highest impact in the `PolicyDocument` domain. It is the only over-represented topic (*i.e.,* $\text{Impact}(t|d) > 1$) across all domains, indicating the significant attention received from a broad range of public audiences on language models.

Observation 2: Papers on ethics, bias, and fairness are highly consumed in policy documents but have significantly lower usage in patents and academic papers.

The topic "*Ethics, Bias, and Fairness*" achieves the highest impact in `PolicyDocument`, exceeding the average in `Media`, while being under-represented in `Citation` and ranking the last in `Patent`. The topic "*Computational Social Science and Cultural Analytics*" exhibits a similar pattern of external impacts, but with a less pronounced contrast.

Observation 3: Linguistic foundations are relatively under-represented in all internal and external domains.

Topics that provide linguistic and theoretical underpinnings of NLP, such as "*Phonology, Morphology, and Word Segmentation*", "*Linguistic Theories, Cognitive Modeling and Psycholinguistics*", as well as "*Discourse and Pragmatics*", have $\text{Impact}(t|d) < 1$ across all domains.

Observation 4: Tasks and techniques benefiting real-world NLP applications have a larger impact in patents.

Besides "*Language Modeling*", topics strongly consumed in `Patent` include "*Machine Learning for NLP*", "*Information Retrieval and Text Mining*", "*Speech Processing and Spoken Language Understanding*", "*Syntax: Tagging, Chunking and Parsing*", as well as "*Machine Translation*". Many of these topics facilitate practical applications, such as search engines and speech recognition systems.

Observation 5: Analyses of the behavior and societal influence of NLP models see stronger adoption in media and policy discussions.

6 topics are over-represented in both `Media` and `PolicyDocument`. Apart from "*Language Model-*

Table 1: Pearson correlation coefficient between the impact of NLP papers in the internal domain (*i.e.,* `Citation`) and that in each external domain.

d	`Patent`	`Media`	`PolicyDocument`
$\mathrm{Corr}(\mathbf{I}_{\texttt{Citation}}, \mathbf{I}_d)$	0.654	0.725	0.247 (0.599 if excluding "*Ethics, Bias, and Fairness*")

ing", the other 5 topics—"*Resources and Evaluation*", "*Interpretability and Analysis of Models for NLP*", "*Ethics, Bias, and Fairness*", "*Multilinguality and Language Diversity*", as well as "*Computational Social Science and Cultural Analytics*"—all address the understanding and evaluation of NLP models.

3.2 Correlation between Internal and External Impacts

Our next study will draw the following conclusion:

Observation 6: NLP papers attracting attention from external domains are more likely to be internally impactful as well.

In Section 3.1, we have raised a few examples whose internal and external impacts are positively correlated. Now, we quantitatively validate this. Intuitively, each domain d can be represented as a vector $\mathbf{I}_d$, where each entry corresponds to the impact of a specific topic within d:

$$\mathbf{I}_d = [\mathrm{Impact}(t|d)]_{t \in \mathcal{T}}.$$

Table 1 presents the Pearson correlation coefficient between $\mathbf{I}_{\texttt{Citation}}$ and each external $\mathbf{I}_d$ ($d \in \{\texttt{Patent}, \texttt{Media}, \texttt{PolicyDocument}\}$). We observe strong positive correlation between `Citation` and `Patent`/`Media` (*i.e.,* with the Pearson correlation coefficient greater than 0.5). The correlation between `Citation` and `PolicyDocument` is weaker, although still positive. That being said, if we remove "*Ethics, Bias, and Fairness*" (which is an evident outlier according to Figure 1) from consideration, $\mathrm{Corr}(\mathbf{I}_{\texttt{Citation}}, \mathbf{I}_{\texttt{PolicyDocument}})$ rises above 0.5 as well.

To present more evidence of the positive correlation between internal and external impacts, we follow Yin et al. (2022) and conduct an experiment that leverages external use of NLP research to predict internally most cited papers. Let us consider the top-1% (internally) highly cited NLP papers in $\mathcal{P}$. If we do not exploit any external signal and randomly pick one paper p from $\mathcal{P}$, the expected probability that p is among the top-1% cited papers (*a.k.a.,* the hit rate) should be 1%. In comparison, Table 2 shows the hit rate if we consider papers referenced at least once in a specific external domain d (*i.e.,* $\#\mathrm{citation}(p|d) \geq 1$). Papers consumed by patents, media, and policy documents exhibit hit rates of 5.46%, 9.26%, and 18.29%, respectively, which are all large multiples of the 1% baseline.

Table 2: Hit rate of predicting the top-1% (internally) highly cited papers when using different external information.

External Domain(s) Considered	**Hit Rate**
$\emptyset$	1.00%
{`Patent`}	5.46%
{`Media`}	9.26%
{`PolicyDocument`}	18.29%
{`Patent, Media`}	26.72%
{`Patent, PolicyDocument`}	34.02%
{`Media, PolicyDocument`}	45.71%
{`Patent, Media, PolicyDocument`}	71.88%

Combining the results from Tables 1 and 2, we find good alignment between what the public from external domains consume and what is regarded as impactful by researchers themselves.

3.3 Complementarity of Different External Impacts

Although we have revealed the positive correlation between `Citation` and each external domain, the alignment between two external domains does not always hold. Qualitatively, we have discussed the different focal points of `Patent` and `Media`/`PolicyDocument` in Observations 4 and 5 in Section 3.1. Quantitatively, we actually have $\mathrm{Corr}(\mathbf{I}_{\texttt{Patent}}, \mathbf{I}_{\texttt{PolicyDocument}}) = -0.140$, implying the complementarity, rather than substitutability, of impacts in patents and policy documents. Table 2 echos this observation. Indeed, for papers referenced in two external domains, the hit rate increases to 26.72%-45.71%, significantly higher than those when we consider a single external domain. A paper referenced in all three external domains is a top-1% cited paper in $\mathcal{P}$ at an astonishing 71.88 times the baseline rate. To summarize, we draw the following conclusion:

Observation 7: Different external domains may favor different types of NLP papers. Papers attracting attention from multiple external domains are more likely to be internally impactful than those attracting one domain only.

4 Conclusions

In this paper, we conduct a large-scale scientometric analysis on the internal and external impacts of NLP papers. We find that the broader academic community and the public's attention to NLP research is primarily driven by the demand for language modeling studies, with the exception of policy-related discussions, which show a greater interest in ethics, bias, and fairness. Although different external domains have varying interests in the specific topics of NLP research, there is an overall positive correlation between the public interests and the impact of papers within the academic "ivory tower". By examining how the broader academic community and the public perceive NLP papers, our analysis offers complementary insights into whether ACL can be considered an AI conference.

Limitations

This work has the following limitations: First, although the three external domains we consider are all important public spaces, they do not encompass all channels that NLP research may impact. Even within these three domains, NLP papers may be consumed through channels not captured by our collected data. Second, we utilize GPT-4o to predict the most relevant topic to each NLP paper. On top of that, we conduct human evaluation for quality assessment. However, there are still chances where GPT-4o's annotations may not be accurate enough, potentially affecting the subsequent analysis. Third, we currently lack results (and data) to explain why there is a positive correlation between internal and external impacts. It might be a possible explanation that either researchers or the public are following the other's preference regarding NLP. Yin et al. (2022) propose another potential hypothesis that each external domain typically involves an "intermediary" to engage with science, such as journalists in the media, inventors in the patent field, and policy experts in the government. They leverage their expertise to select scientific results and introduce them into their respective domains. Finally, since our collected data from certain domains do not indicate when an NLP paper was cited, we are unable to perform a systematic temporal analysis. We believe that, with appropriate data support, all of the limitations mentioned above represent highly promising future directions.

Acknowledgments

We thank Yian Yin for helpful suggestions.

References

Euan Adie and William Roe. 2013. Altmetric: enriching scholarly content with article-level discussion and metrics. *Learned Publishing*, 26(1):11–17.

Hancheng Cao, Yujie Lu, Yuting Deng, Daniel Mcfarland, and Michael S Bernstein. 2023. Breaking out of the ivory tower: A large-scale analysis of patent citations to hci research. In *CHI'23*, pages 760:1–760:24.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. In *NAACL-HLT'19*, pages 3586–3596.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. In *ACL'20*, pages 2270–2282.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *TACL*, 6:391–406.

Matt Marx and Aaron Fuegi. 2020. Reliance on science: Worldwide front-page patent citations to scientific articles. *Strategic Management Journal*, 41(9):1572–1594.

Matt Marx and Aaron Fuegi. 2022. Reliance on science by inventors: Hybrid extraction of in-text patent-to-article citations. *Journal of Economics & Management Strategy*, 31(2):369–392.

Jason Priem, Heather Piwowar, and Richard Orr. 2022. Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. *arXiv preprint arXiv:2205.01833*.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *WWW'15*, pages 243–246.

Martin Szomszor and Euan Adie. 2022. Overton: A bibliometric database of policy document citations. *Quantitative Science Studies*, 3(3):624–650.

Yian Yin, Yuxiao Dong, Kuansan Wang, Dashun Wang, and Benjamin F Jones. 2022. Public use and public funding of science. *Nature Human Behaviour*, 6(10):1344–1350.

Yian Yin, Jian Gao, Benjamin F Jones, and Dashun Wang. 2021. Coevolution of policy and science during the pandemic. *Science*, 371(6525):128–130.

Yu Zhang, Xiusi Chen, Bowen Jin, Sheng Wang, Shuiwang Ji, Wei Wang, and Jiawei Han. 2024. A comprehensive survey of scientific large language models and their applications in scientific discovery. In *EMNLP'24*, pages 8783–8817.

A Topic Prediction of NLP Papers

A.1 Topic Annotation using GPT-4o

We adopt the following instruction prompt to guide GPT-4o in performing topic annotation.

```
Instruction: You will be given the title and abstract of a natural language processing paper, as well as a list of candidate topics. Select the most relevant topic to the given paper from the list.

Paper title: [title; e.g., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding]

Paper abstract: [abstract; e.g., We introduce a new language representation model called BERT, which stands for ...]

Candidate labels: "Computational Social Science and Cultural Analytics", "Dialogue and Interactive Systems", "Discourse and Pragmatics", "Ethics, Bias, and Fairness", ... (25 topics)
```

We set the temperature as 1.0 (the default setting for GPT-4o in OpenAI's API). In rare cases, the output of GPT-4o does not exactly match any of the candidate labels. To tackle this, we use a scientific pretrained language model, SPECTER (Cohan et al., 2020), to encode the output and each candidate label's name; then, we consider the nearest neighbor of the output in the SPECTER embedding space as the predicted label.

A.2 Human Evaluation

To assess the quality of annotations produced by GPT-4o, human evaluation is then conducted. We recruit 3 evaluators, all of whom have rich NLP research experiences. We randomly selected 100 samples for evaluation. For each submission, we ask each annotator to judge the correctness of the label predicted by GPT-4o. The scoring scale is 2 (the predicted label is the most relevant label to the paper), 1 (the predicted label is relevant to the paper, although there is a better option), and 0 (the predicted label is irrelevant to the paper).

According to human evaluation, GPT-4o gets an average score of 1.53 across the three evaluators. Notably, none of the 100 predictions gets 0 (*i.e.,* irrelevant), indicating the reliability of GPT-4o annotations. We also calculate the Fleiss' kappa (Fleiss, 1971), which is 0.62 and implies "substantial agreement" among the evaluators.

B Impact of NLP Papers in GitHub

As noted in the Limitations section, the impact of an NLP paper may not be limited to textual citations alone. As a complementary analysis, in this section, we examine the influence of NLP papers in the `GitHub` domain. Specifically, we analyze the average number of forks for GitHub repositories associated with each NLP topic. It is important to note that we consider this measure of impact to be a mixture of both internal and external influences, as forking a code repository may be done by other researchers for follow-up research, or by practitioners aiming to deploy the work in real-world applications.

B.1 Data and Metric

We refer to Papers With Code[9] to establish the correspondence between our collected NLP papers and their associated GitHub repositories. 7,113 (out of 21,104) NLP papers are mapped to at least one repository in this way. Then, we used the GitHub API[10] to retrieve the number of forks for each corresponding repository. If a paper p is associated with multiple repositories, its fork count $\#\text{fork}(p)$ is computed as the sum of forks across all associated repositories. Analogous to defining impact through the citation count, we define the impact of NLP papers in `GitHub` based on the fork count as follows:

$$\text{Impact}(t|\texttt{GitHub}) = \frac{\sum_{p \in \mathcal{P}_t} \#\text{fork}(p)/|\mathcal{P}_t|}{\sum_{p \in \mathcal{P}} \#\text{fork}(p)/|\mathcal{P}|}.$$

Here, we only include NLP papers in $\mathcal{P}$ and $\mathcal{P}_t$ that can be matched to at least one GitHub repository.

B.2 Analysis

Figure A1 shows $\text{Impact}(t|\texttt{GitHub})$ across all topics. Once again, we observe that "*Language Modeling*" exhibits the highest impact by a significant margin. Other topics with relatively high impact include "*Speech Processing and Spoken Language Understanding*", "*Machine Learning for NLP*",

[9] https://production-media.paperswithcode.com/about/links-between-papers-and-code.json.gz, accessed on March 5, 2025.

[10] https://api.github.com/repos, accessed on March 5, 2025.

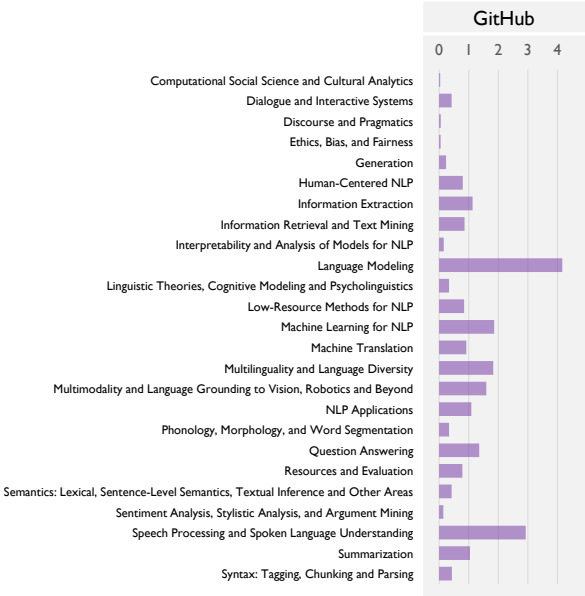


Figure A1: Impact of NLP papers with a certain topic in the `GitHub` domain.

Table A1: Pearson correlation coefficient between the impact of NLP papers in `GitHub` and that in other domains.

d	`Citation`	`Patent`	`Media`	`Policy Document`
$\mathrm{Corr}(\mathbf{I}_{\texttt{GitHub}}, \mathbf{I}_d)$	0.586	0.633	0.531	0.009

"*Multilinguality and Language Diversity*", as well as "*Multimodality and Language Grounding to Vision, Robotics and Beyond*". It is evident that topics related to practical NLP applications tend to achieve higher impact in `GitHub`. In this regard, `GitHub` is similar to `Patent`, which also aligns with our intuition. By contrast, linguistic foundations and "*Ethics, Bias, and Fairness*" exhibit significantly lower impact.

Table A1 demonstrates the Pearson correlation coefficient between the impact of NLP papers in `GitHub` and that in other domains. We observe a clear positive correlation between `GitHub` and `Citation`, `Patent`, and `Media`, with the strongest correlation observed with `Patent`, echoing our qualitative discussion above. Meanwhile, the near-zero correlation between `GitHub` and `PolicyDocument` reflects their distinct emphases.

An Effective Incorporating Heterogeneous Knowledge Curriculum Learning for Sequence Labeling

Xuemei Tang[1], Jun Wang[2], Qi Su[3]*, Chu-Ren Huang[1], and Jinghang Gu[1]*
[1]The Hong Kong Polytechnic University, Hong Kong, China
[2]Department of Information Management, Peking University, Beijing, China
[3]School of Foreign Languages, Peking University, Beijing, China
xuemeitang00@gmail.com, {sukia,junwang}@pku.edu.cn
{churen.huang, jinghang.gu}@polyu.edu.hk

Abstract

Sequence labeling models often benefit from incorporating external knowledge. However, this practice introduces data heterogeneity and complicates the model with additional modules, leading to increased expenses for training a high-performing model. To address this challenge, we propose a dual-stage curriculum learning (DCL) framework specifically designed for sequence labeling tasks. The DCL framework enhances training by gradually introducing data instances from easy to hard. Additionally, we introduce a dynamic metric for evaluating the difficulty levels of sequence labeling tasks. Experiments on several sequence labeling datasets show that our model enhances performance and accelerates training, mitigating the slow training issue of complex models [1].

1 Introduction and Related Work

Sequence labeling is a core task in natural language processing (NLP) that involves assigning labels to individual elements in a sequence. Recent advancements in neural network methods have significantly improved performance in sequence labeling tasks (Zhang et al., 2014; Chen et al., 2017a; Zhang et al., 2018; Tian et al., 2020a; Nguyen et al., 2021; Hou et al., 2021; Liu et al., 2021). Some studies have explored integrating external knowledge, such as n-grams, lexicons, and syntax, to enhance these models. However, this integration adds heterogeneity and complexity to the input data. Additionally, incorporating such knowledge often necessitates extra encoding modules, like attention mechanisms (Liu et al., 2021; Tian et al., 2020b) or graph neural networks (GNN) (Chen et al., 2017b; Gui et al., 2019; Nie et al., 2022), which increase model parameters and make the system more computationally expensive to develop.

Curriculum Learning (CL) (Bengio et al., 2009) effectively addresses these challenges by simulating the human learning process, where training samples are introduced progressively from easy to hard. This approach facilitates efficient learning from heterogeneous data while enhancing both the speed and performance of the model (Bengio et al., 2009; Wang et al., 2021). CL has shown success in a variety of NLP tasks, including machine translation (Wan et al., 2020), dialogue generation (Zhu et al., 2021), and text classification (Zhang et al., 2022). Data-selection strategies are crucial in CL. However, these difficulty metrics primarily focus on the sentence level, such as Mohiuddin et al. (2022), Yuan et al. (2022) and Liu et al. (2024)'s works, and there is a lack of token-level and word-level metrics to measure the difficulty of sequence labeling tasks.

To address this gap, in this paper, we introduce a dual-stage curriculum learning (DCL) framework specifically designed for sequence labeling tasks. The first stage is data-level CL, where we train a basic teacher model on all available training data, aiming to alleviate the cold start problem of the student model. The second stage is model-level CL, where we start training the student model on a selected subset of the teacher model and gradually expand the training subset by considering the difficulty of the data and the state of the student model. Furthermore, we explore different difficulty metrics for sequence labeling tasks within the DCL framework. These metrics include a pre-defined metric, such as sentence length, and model-aware metrics, namely Top-N least confidence (TLC), Maximum normalized log-probability (MNLP), and Bayesian uncertainty (BU). Finally, we choose the classical sequence labeling tasks, Chinese word segmentation (CWS), part-of-speech (POS) tagging, and named entity recognition (NER), to validate our proposed approach.

*Corresponding authors.
[1]https://github.com/tangxuemei1995/DCL4SeqLabeling

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 495–503
July 27 - August 1, 2025

2 Method

The framework proposed in this study consists of three main components: a teacher sequence labeling model, a student sequence labeling model, and a DCL training strategy. It is worth noting that the DCL is independent of the sequence labeling model.

Following previous works (Zhang et al., 2018; Gong et al., 2019; Fu et al., 2020), in sequence labeling tasks, we feed an input sentence $X = \{x_1, ...x_i..., x_M\}$ into the encoder, and the decoder then outputs a label sequence $Y^* = \{y_1^*, ...y_i^*...y_M^*\}$, where y_i^* represents a label from a pre-defined label set T, and M denotes the length of sentence.

2.1 Dual-stage Curriculum Learning

Algorithm 1 Training Process with DCL

Input: Original corpus $\mathcal{D}$, difficulty metric $S(\cdot)$, teacher model epochs E_0, student model epochs E_s, scheduler λ, length function $|\cdot|$
Output: Trained student model θ
// Data-level Curriculum Learning
1: Train teacher model θ_0 on $\mathcal{D}$ for E_0 epochs
2: Compute $S(\theta_0)$ for each sample in $\mathcal{D}$
3: Sort $\mathcal{D}$ by $S(\theta_0)$ in ascending order to obtain ranked dataset $\mathcal{D}_r$
// Model-level Curriculum Learning
4: Initialize λ_0 (starting curriculum ratio)
5: $m \leftarrow \lambda_0 \cdot |\mathcal{D}|$
6: Student training set $\mathcal{D}_s \leftarrow \mathcal{D}_r[0:m]$
7: Remaining data $\mathcal{D}_o \leftarrow \mathcal{D}_r[m:]$
8: **for** *epoch* $= 1$ to E_s **do**
9: **if** $\lambda < 1$ **then**
10: *a)* Train student model on $\mathcal{D}_s$ to obtain current θ_*
11: *b)* Compute $S(\theta_*)$ for all samples in $\mathcal{D}_o$
12: *c)* Sort $\mathcal{D}_o$ by $S(\theta_*)$ in ascending order to get updated $\mathcal{D}_r$
13: *d)* Update λ using Eq. 6
14: *e)* Calculate new data size: $m \leftarrow \lambda \cdot |\mathcal{D}| - |\mathcal{D}_s|$
15: *f)* Expand $\mathcal{D}_s$ with new samples: $\mathcal{D}_s \mathrel{+}= \mathcal{D}_r[0:m]$
16: *g)* Update remaining data: $\mathcal{D}_o \leftarrow \mathcal{D}_r[m:]$
17: **else**
18: Train student model on $\mathcal{D}_s$
19: **end if**
20: **end for**

We propose a novel dual-stage curriculum learning approach: *data-level* CL and *model-level* CL, as detailed in Algorithm 1.

At the data level, we first train a basic teacher model on the entire dataset $\mathcal{D}$ for E_0 epochs, where E_0 is smaller than the total epochs needed for convergence (Line 1). The teacher model θ_0 is then used to calculate difficulty scores $S(\theta_0)$ for each sample (Line 2), and the samples are sorted by difficulty to form a ranked dataset $\mathcal{D}_r$ (Line 3).

At the model level, we address the cold-start issue by initializing the student model training set $\mathcal{D}_s$ with a subset of $\mathcal{D}_r$ (Lines 4-6). The proportion of samples, controlled by the parameter λ, governs the curriculum learning process. The remaining data, $\mathcal{D}_o$, is incorporated into $\mathcal{D}_s$ gradually as λ increases. The number of new samples to be added is denoted as m (Lines 5, 14).

The student model is trained on $\mathcal{D}_s$ to update the model parameters θ_* (Line 10). Then, θ_* is used for the difficulty calculation of the samples in $\mathcal{D}_o$ (Line 11). Next, $\mathcal{D}_o$ is ranked by new difficulty scores, forming a new ranked dataset $\mathcal{D}_r$ (Line 12). The threshold λ is updated (Line 13), and new samples are added to $\mathcal{D}_s$ based on λ (Lines 14-15). As λ approaches 1, all of $\mathcal{D}_o$ is added to $\mathcal{D}_s$. The complete dataset is then used to train the student model to convergence.

The key elements in Algorithm 1 are the difficulty metric $S(\cdot)$ and threshold λ, which control the difficulty ranking of samples and the progression of training, respectively. The design of these components will be discussed in the following sections.

2.2 Difficulty Metrics

We now provide a detailed formulation for calculating the difficulty $S(\cdot)$ in Algorithm 1. In sequence labeling tasks, sample difficulty is tied to individual tokens, but assessing token-level difficulty is challenging. We use uncertainty from active learning to measure the model's confidence in labeling training samples.

Bayesian Uncertainty (BU). Following Buntine and Weigend (1991), model uncertainty can be assessed using Bayesian Neural Networks. As noted by Wang et al. (2019), higher predicted probability variance indicates greater uncertainty, suggesting that the model is less confident about the sample. In this work, we employ the widely-used Monte Carlo dropout (Gal and Ghahramani, 2016) to approximate Bayesian inference.

First, we apply Monte Carlo dropout (Gal and Ghahramani, 2016) to obtain each sample of token-level tagging probabilities. Specifically, for each token x_i, we perform K stochastic forward passes through the model, resulting in K predicted distributions $P(y_i \mid x_i)_1, \ldots, P(y_i \mid x_i)_K$. This provides K predictions with associated probabilities for each token. Then the expectation of token-level

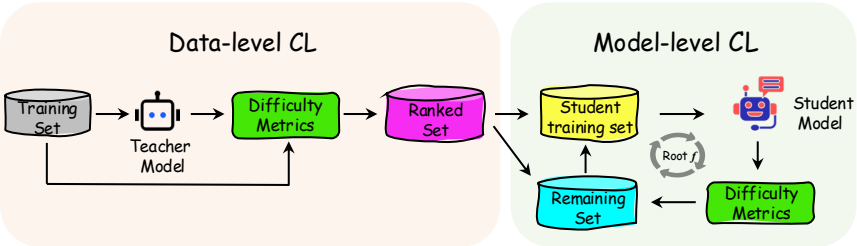


Figure 1: The framework of the proposed model consists of a teacher model, a student model, and a DCL strategy. Here, "Root f" represents *Root function*.

tagging probability can be approximated by

$$\mathbb{E}[P(y_i|x_i)] \approx \frac{1}{K}\sum_{k=1}^{K} P(y_i|x_i)_k \quad (1)$$

The variance of token-level tagging probability on the label set can be approximated by

$$var(x_i,\theta) \approx \sum_{y_i \in T}(\frac{1}{K}\sum_{k=1}^{K} P(y_i|x_i)_k^2 - \mathbb{E}[P(y_i|x_i)]^2) \quad (2)$$

Now, we obtain the variance of each token $var(x_i,\theta)$, then we use the average variance score of all tokens in the sequence as the sentence-level variance as follows.

$$var(\theta)_{aver.} = \frac{1}{M}\sum_{i=1}^{M} var(x_i,\theta) \quad (3)$$

The maximum variance score $var(\theta)_{max}$ also is valuable, which reflects the highest uncertainty in the sequence.

$$var(\theta)_{max} = max_{i\in[1,M]} var(x_i,\theta) \quad (4)$$

The final uncertainty score or difficulty score of each sequence is calculated as follows.

$$S(\theta)^{BU} = var(\theta)_{max} + var(\theta)_{aver.} \quad (5)$$

Both at the data level and model level, the difficulty of training samples is measured by the above various $S(\theta)$.

2.3 Training Scheduler

The training scheduler regulates the pace of CL. In our approach, we employ the *Root function* as the control mechanism. This function ensures that the model receives sufficient time to learn newly introduced examples while gradually decreasing the number of newly added examples throughout the training process.

$$\lambda = \min\left(1, \sqrt{\frac{1-\lambda_0^2}{E_{grow}} \cdot t + \lambda_0^2}\right) \quad (6)$$

where E_{grow} denotes the number of epochs required for λ to reach 1, while $\lambda_0 > 0$ represents the initial proportion of the easiest training samples. t indicates the t_{th} training epochs. When λ reaches 1, the model has access to the entire training dataset.

3 Experiments

3.1 Dataset and Experimental Configurations

Dataset. Chinese word segmentation (CWS) and part-of-speech (POS) tagging are representative sequence labeling tasks. So we evaluate our approach using three CWS and POS tagging datasets, including Chinese Penn Treebank version 5.0 [2], 6.0 [3], and PKU. More dataset details can be found in Appendix A.

Teacher and student models. In this study, the basic transfer teacher framework is RoBERTa + Softmax. For the student model, we select two representative complex models introduced by Tian et al. (2020b) and Tang et al. (2024). In their work, Tian et al. (2020b) employed an attention mechanism framework, McASP, to integrate lexicons and n-grams for the joint CWS and POS tagging task, using BERT as the encoder. Meanwhile, Tang et al. (2024) incorporated syntax and semantic knowledge into sequence labeling tasks through a GCN framework called SynSemGCN, with RoBERTa as the sequence encoder.

Curriculum learning baselines. We compare our difficulty metric with four baseline difficulty metrics for CL: ***a. Random***: Samples are assigned in random order; ***b. Sentence Length (Length)***: Samples are ranked from shortest to longest, based on the intuition that longer sequences are more challenging to encode; (Random and Length metics represent simple CL, namely without the teacher model). ***c. Top-N Least Confidence (TLC)***: The difficulty of a sequence is determined by using the

[2] https://catalog.ldc.upenn.edu/LDC2005T01
[3] https://catalog.ldc.upenn.edu/LDC2007T36

Model	CL Setting	CTB5		CTB6		PKU	
		CWS	POS	CWS	POS	CWS	POS
Tian et al. (2020a)	-	98.73	96.60	97.30	94.74	-	-
Tian et al. (2020b) (McASP)	-	98.77	96.77	97.43	94.82	-	-
Liu et al. (2021)	-	-	97.14	-	-	-	-
Tang et al. (2024) (SynSemGCN)	-	98.83	96.77	97.86	94.98	98.05	95.50
McASP	Rand.	98.81	96.84	97.37	94.90	98.38	96.27
	Length	98.83	96.85	97.35	94.82	98.40	96.25
	TLC	98.83	96.89	97.37	94.83	98.41	96.30
	MNLP	98.85	96.81	97.41	94.92	98.41	96.30
	BU	**98.91**	96.87	97.42	94.90	98.43	96.32
SynSemGCN	Rand.	98.84	97.86	97.99	95.05	98.48	96.40
	Length	98.80	96.84	97.40	94.94	98.53	96.48
	TLC	98.83	97.81	97.98	95.02	**98.61**	**96.55**
	MNLP	98.78	97.72	98.04	95.13	98.56	96.48
	BU	98.90	**97.95**	**98.05**	**95.14**	98.59	96.54

Table 1: Experimental results of different models using different CL settings on test sets of three datasets. Here, "CWS" represents the F1 value of CWS, and "POS" means the F1 value of the joint CWS and POS tagging. "-" means without the CL training strategy, and "TLC", "MNLP", and "BU" means using the DCL setting with different difficulty metrics. The maximum F1 scores for each dataset are highlighted.

N tokens with the lowest confidence; ***d. Maximum Normalized Log-Probability (MNLP)***: The difficulty is assessed by calculating the product of the label probabilities for all tokens in the sequence. The detailed computation processes for TLC and MNLP are provided in Appendix B.

For further details on the important hyper-parameters of the model, please refer to Appendix C. We discuss the selection process of these parameter values in detail in the Appendix D.

3.2 Overall Experimental Results

Table 1 presents the experimental results of baselines and two models with different CL settings. The experimental results reveal several noteworthy conclusions.

Firstly, the DCL methodology introduced in this paper is flexible and can be integrated with various complex models. As shown in Table 1, the difficulty metrics proposed here outperform the Random and Length metrics across most datasets. Specifically, the BU metric consistently delivers the best performance on the majority of datasets when applied to the SynSemGCN model, surpassing the TLC and MNLP metrics.

Additionally, we compare our approach with previous methods that incorporate external knowledge or resources into the encoder. The results reveal that models using CL exhibit significant performance improvements, surpassing the performance of earlier methods.

Model	CTB5		Time
	CWS	POS	
Ours	**98.90**	**97.95**	287m
w/o data CL(BU)	**98.90**	97.88	-
w/o model CL(BU)	98.85	97.51	-
w/o DCL	98.75	96.73	393m

Table 2: Ablation experimental results of DCL. The baseline model "w/o DCL" denotes the model SynSemGCN; "w/o model CL" means the student model always uses the initial data order sorted by the transfer teacher model; "w/o data CL" indicates the initial training samples for the student model is drawn randomly from the training set; "Ours" indicates "SynSemGCN+DCL(BU)". Both teacher and student models with DCL in this table use BU as the difficulty metric. "Time" means the training time (in minutes).

Model	CTB5	
	CWS	POS
McASP with BU	**98.91**	**96.87**
w/o $var(\theta)_{max}$	98.78	96.78
w/o $var(\theta)_{aver.}$	98.86	96.74
McASP	98.73	96.60

Table 3: Ablation experimental results of two parts in BU metrics (Eq. 5).

Models	Weibo (Chinese)	Note4 (Chinese)	CoNLL-2003 (English)
BERT	66.22	79.15	90.94
BERT + CL (Length)	66.81	79.63	90.79
BERT + DCL (TLC)	**67.52**	79.53	91.30
BERT + DCL (MNLP)	65.73	79.95	91.15
BERT + DCL (BU)	66.74	**80.02**	**91.77**

Table 4: Performance comparison of different difficulty metrics on three NER datasets.

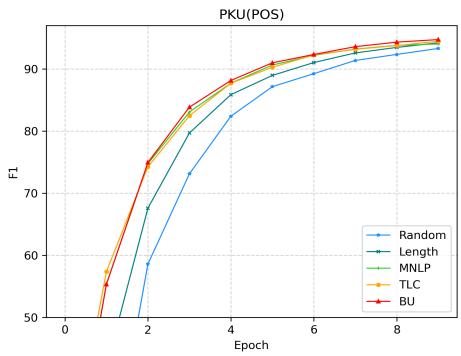


Figure 2: The F1 scores on the dev set of PKU with different difficulty metrics in the model-level CL training process.

3.3 Effect of Dual-stage Curriculum Learning

In this section, we discuss the impact of DCL. We perform ablation studies by removing either the data-level CL or the model-level CL. The results are summarized in Table 2. Model-level CL has a more significant impact than data-level CL. This is intuitive, as model-level CL influences the entire training process, while data-level CL primarily affects the early stages of student model training.

We also compare the training time of models with and without DCL. The experimental results in Table 2 show that all models were trained for 50 epochs. The training time for models using DCL includes the time spent on training the teacher model and calculating the difficulty values for the student model. The results indicate that DCL improves model performance and reduces training time by over 25%.

3.4 Ablation Study on BU Difficulty Components

We adopt McASP (Tian et al., 2020b) as the backbone model, incorporating DCL as the training strategy and BU as the difficulty metric. To examine the contribution of each component in BU, we conduct ablation experiments on its two parts: $var(\theta)_{max}$ and $var(\theta)_{aver.}$, as shown in Table 3. Removing either component results in performance degradation, indicating that both components are crucial. Moreover, the comparable drop in performance suggests that $var(\theta)_{max}$ and $var(\theta)_{aver.}$ contribute similarly to the effectiveness of DCL.

3.5 Comparison of Difficulty Metrics

In this section, we examine the impact of different difficulty metrics during the model-level CL training process for the SynSemGCN model. Figure 2 shows the F1 score change on the PKU dataset development set over the first 10 epochs of model-level CL training. After 10 epochs, all training data are used, so the initial 10 epochs highlight the effect of different metrics. From the figure, we observe that BU, in particular, achieves the best performance, indicating that uncertainty-based metrics can select samples that better align with the model's learning trajectory, leading to faster learning.

3.6 Generalization Capability

We conduct additional experiments to demonstrate the applicability of our method to the NER task. We select two Chinese and one English NER datasets: Weibo [4], OntoNotes4 [5] and CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003). The statistics of three datasets are shown in Table 6. We compare the performance of models using DCL and CL (Length) with a model without CL on these datasets. As shown in Table 4, the results of the DCL method outperform those of BERT+CL (Length) and BERT (no CL), indicating the effectiveness of our method. This also suggests that our method can be applied to sequence labeling tasks beyond CWS and POS tagging.

4 Conclusion

This paper introduces a novel dual-stage curriculum learning framework aimed at enhancing performance and accelerating the training process for sequence labeling tasks. Focusing on the sequence labeling task of CWS, POS tagging, and NER, this framework demonstrates its effectiveness.

[4] https://catalog.ldc.upenn.edu/LDC2013T19/
[5] https://github.com/cchen-nlp/weiboNER

Limitations

There are several limitations to our study. First, the design of our difficulty metrics involves the tuning of multiple hyperparameters, which may complicate optimization. Second, we did not explore a curriculum learning process that progresses from hard to easy examples. Third, we focused on a single variation of the λ parameter to control CL and did not investigate alternative methods for adding training data.

Acknowledgments

This research is supported by the NSFC project "The Construction of the Knowledge Graph for the History of Chinese Confucianism" (Grant No. 72010107003) and The Hong Kong Polytechnic University Project "Evaluating the Syntax-Semantics Knowledge in Large Language Models" (Grant No. P0055270).

References

Ankit Agrawal, Sarsij Tripathi, and Manu Vardhan. 2021. Active learning approach using a modified least confidence sampling strategy for named entity recognition. *Progress in Artificial Intelligence*, 10(2):113–128.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Wray L. Buntine and Andreas S. Weigend. 1991. Bayesian back-propagation. *Complex Systems*.

Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2017a. A feature-enriched neural model for joint chinese word segmentation and part-of-speech tagging. In *IJCAI*.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017b. Dag-based long short-term memory for neural word segmentation. *Preprint*, arXiv:1707.00248.

Aron Culotta and Andrew McCallum. 2005. *Reducing Labeling Effort for Structured Prediction Tasks:*. Fort Belvoir, VA.

Jinlan Fu, Pengfei Liu, Qi Zhang, and Xuanjing Huang. 2020. Rethink cws: Is chinese word segmentation a solved task? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5676–5686, Online. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. (arXiv:1506.02142). ArXiv:1506.02142 [cs, stat].

Jingjing Gong, Xinchi Chen, Tao Gui, and Xipeng Qiu. 2019. Switch-lstms for multi-criteria chinese word segmentation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6457–6464. AAAI Press.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuanjing Huang. 2019. A lexicon-based graph neural network for chinese ner. page 11.

Yang Hou, Houquan Zhou, Zhenghua Li, Yu Zhang, Min Zhang, Zhefeng Wang, Baoxing Huai, and Nicholas Jing Yuan. 2021. A coarse-to-fine labeling framework for joint word segmentation, pos tagging, and constituent parsing. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, page 290–299, Online. Association for Computational Linguistics.

Honglin Liu, Peng Hu, Changqing Zhang, Yunfan Li, and Xi Peng. 2024. Interactive deep clustering via value mining. In *Advances in Neural Information Processing Systems*, volume 37, pages 42369–42387. Curran Associates, Inc.

Wei Liu, Xiyan Fu, Yue Zhang, and Wenming Xiao. 2021. Lexicon enhanced chinese sequence labeling using bert adapter. *arXiv:2105.07148 [cs]*. ArXiv: 2105.07148.

Tasnim Mohiuddin, Philipp Koehn, Vishrav Chaudhary, James Cross, Shruti Bhosale, and Shafiq Joty. 2022. Data selection curriculum for neural machine translation. (arXiv:2203.13867). ArXiv:2203.13867 [cs].

Duc-Vu Nguyen, Linh-Bao Vo, Ngoc-Linh Tran, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. 2021. Joint chinese word segmentation and part-of-speech tagging via two-stage span labeling. *arXiv:2112.09488 [cs]*. ArXiv: 2112.09488.

Yu Nie, Yilai Zhang, Yongkang Peng, and Lisha Yang. 2022. Borrowing wisdom from world: modeling rich external knowledge for chinese named entity recognition. *Neural Computing and Applications*, 34(6):4905–4922.

Yan Shao, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2017. Character-based joint segmentation and pos tagging for chinese using bidirectional rnn-crf. *Preprint*, arXiv:1704.01314.

Yanyao Shen, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, and Animashree Anandkumar. 2018. Deep active learning for named entity recognition. (arXiv:1707.05928). ArXiv:1707.05928 [cs].

Xuemei Tang, Jun Wang, and Qi Su. 2024. Incorporating knowledge for joint chinese word segmentation and part-of-speech tagging with synsemgcn. *Aslib Journal of Information Management*, ahead-of-print(ahead-of-print).

Yuanhe Tian, Yan Song, Xiang Ao, Fei Xia, Xiaojun Quan, Tong Zhang, and Yonggang Wang. 2020a. Joint Chinese word segmentation and part-of-speech tagging via two-way attentions of auto-analyzed knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8286–8296, Online. Association for Computational Linguistics.

Yuanhe Tian, Yan Song, and Fei Xia. 2020b. Joint Chinese word segmentation and part-of-speech tagging via multi-channel attention of character n-grams. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2073–2084, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Yu Wan, Baosong Yang, Derek F. Wong, Yikai Zhou, Lidia S. Chao, Haibo Zhang, and Boxing Chen. 2020. Self-paced learning for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 1074–1080, Online. Association for Computational Linguistics.

Kun Wang, Chengqing Zong, and Keh-Yih Su. 2010. A character-based joint model for chinese word segmentation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, page 1173–1181, Beijing, China. Coling 2010 Organizing Committee.

Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019. Improving back-translation with uncertainty-based confidence estimation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, page 791–802, Hong Kong, China. Association for Computational Linguistics.

Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. (arXiv:2010.13166). ArXiv:2010.13166 [cs].

Siyu Yuan, Deqing Yang, Jiaqing Liang, Zhixu Li, Jinxi Liu, Jingyue Huang, and Yanghua Xiao. 2022. Generative entity typing with curriculum learning. (arXiv:2210.02914). ArXiv:2210.02914 [cs].

Meishan Zhang, Nan Yu, and Guohong Fu. 2018. A simple and effective neural model for joint word segmentation and pos tagging. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(9):1528–1538.

Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2014. Type-supervised domain adaptation for joint segmentation and POS-tagging. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 588–597, Gothenburg, Sweden. Association for Computational Linguistics.

Xulong Zhang, Jianzong Wang, Ning Cheng, and Jing Xiao. 2022. Improving imbalanced text classification with dynamic curriculum learning. (arXiv:2210.14724). ArXiv:2210.14724 [cs].

Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting bleu/nist scores: How much improvement do we need to have a better system? In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Qingqing Zhu, Xiuying Chen, Pengfei Wu, JunFei Liu, and Dongyan Zhao. 2021. Combining curriculum learning and knowledge distillation for dialogue generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, page 1284–1295, Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Dataset

The details of the three datasets are given in Table 5. Regarding the CTB datasets, we follow the same approach as previous works (Shao et al., 2017; Tian et al., 2020a) by splitting the data into train/dev/test sets. In the case of PKU, We randomly select 10% of the training data to create the development set.

Datasets		CTB5	CTB6	PKU
Train	Sent.	18k	23K	17k
	Word	494k	99k	482k
Dev	Sent.	350	2K	1.9k
	Word	7k	60K	53k
Test	Sent.	348	3K	3.6k
	Word	8k	12k	97k

Table 5: Detail of the three datasets.

Datasets	Type	Train	Dev	Test
Weibo	Sentences	1.35K	0.27K	0.27K
	Entities	1.89K	0.39K	0.42K
OntoNotes	Sentences	15.7K	4.3K	4.3K
	Entities	13.4K	6.95K	7.7K
CoNLL2003	Sentences	15.0K	3.5K	3.7K
	Entities	23.5K	5.9K	5.7K

Table 6: Detail of the two NER datasets.

B Difficulty Metric Baselines

Top-N least confidence (TLC). Culotta and McCallum (2005) proposed a confidence-based strat-

egy for sequence models called least confidence (LC). This approach sorts the samples in ascending order based on the probability of the most possible label predicted by the model.

The least confidence of each token is calculated as follows.

$$\phi^{LC}_{(x_i,\theta)} = 1 - max_{y_i \in T} P(y_i|x_i) \quad (7)$$

where x_i is the i_{th} token in the input sentence, θ denotes model parameters, y_i is a pre-defined label, T represents the pre-defined label set. $max_{y_i \in T} P(y_i|x_i)$ aims to find the probability of the most possible label predicted by the model. The smaller $\phi^{LC}_{(x_i,\theta)}$ reflects the more confident the model is in predicting the label of x_i.

According to Agrawal et al. (2021), the confidence level of a sentence in a sequence labeling task is typically determined based on a set of representative tokens. Therefore, we select the top N tokens with the highest least confidence in the sentence and then use their average value as the difficulty score of the sentence. Finally, the TLC difficulty metric is formulated as follows.

$$S(\theta)^{TLC} = \frac{1}{N} \sum_{n=1}^{N} \phi^{LC}_{(x_n,\theta)} \quad (8)$$

Maximum normalized log-probability (MNLP). Shen et al. (2018) used MNLP as a confidence strategy to find the product of the maximum probabilities of each token, which is equivalent to taking the logarithm of each probability and summing them. Finally, it is normalized to obtain the confidence score of the sentence as follows.

$$\prod_{i=1}^{M} max_{y_i \in T} P(y_i|x_i) \Longleftrightarrow \sum_{i=1}^{M} log\{max_{y_i \in T} P(y_i|x_i)\} \quad (9)$$

where M is the length of the sentence. The difficulty of a sentence decreases as the confidence level increases. To account for this relationship, we introduce a negative sign. Additionally, in order to reduce the impact of sentence length, we apply a normalization operation. Finally, MNLP is formulated as follows.

$$S(\theta)^{MNLP} = -\frac{1}{M} \sum_{i=1}^{M} log\{max_{y_i \in T} P(y_i|x_i)\} \quad (10)$$

C Parameters Setting

The key experimental parameter settings are shown in Table 7.

Hyper-parameters	Value
E_0	5
E_s	50
λ_0	0.3
E_{grow}	10
K	3
N	5

Table 7: Experiment hyper-parameters setting.

E_{grow}	CTB5		PKU	
	CWS	POS	CWS	POS
5	98.88	97.89	98.65	**97.01**
10	**99.06**	**98.96**	**98.77**	96.97
15	98.84	97.69	98.70	96.90

Table 8: The effect of E_{grow} in Eq. 6.

D Effect of Hyper-parameters

In this section, we explore the impact of the hyperparameters on the performance of DCL. The adjustment of the parameters is based on the SynSemGCN+DCL(BU) model.

First, we investigate the impact of the hyperparameter λ_0 on DCL performance. We conduct the experiments on the CTB5 dataset, tuning the value of λ_0 in the model-level pacing function Eq. 6, and the experimental results are represented by a line graph as shown in Figure 3. As observed, the model achieves optimal performance when $\lambda_0 = 0.3$. However, when the value exceeds 0.4, the model's performance gradually deteriorates.

Additionally, we examine the impact of E_{grow} in Eq. 6, which controls the number of epochs for λ to reach 1. As shown in Table 8, when E_{grow} is set to 10, the model exhibits superior performance on both the CTB5 and PKU datasets. Therefore, we adopt E_{grow} as 10 epochs in our experiments.

Next, we assess the impact of the training epochs E_0 of the teacher model, which initializes the difficulty ranking of the training data for the student model. We aim to investigate whether a more mature teacher model contributes to improved performance. For this purpose, we conduct experiments on both the CTB5 and PKU datasets, utilizing teacher models trained for 5, 10, and 15 epochs

E_0	CTB5		PKU	
	CWS	POS	CWS	POS
5	**99.06**	**98.96**	**98.77**	**96.97**
10	98.98	97.90	98.54	96.69
15	98.73	96.87	98.53	96.54

Table 9: The impact of the number of epochs of teacher model, E_0.

Para.	CTB6		PKU	
	CWS	POS	CWS	POS
K=2	**98.17**	95.43	98.73	96.58
K=3	98.10	**95.59**	**98.77**	**96.97**
K=4	98.09	95.56	98.69	96.38

Table 10: The effect of of K times dropout in BU difficulty metric.

to rank the initial training data for the student models.

The experimental results, as shown in Table 9, reveal that a more mature teacher model does not necessarily lead to better performance. Instead, the student model achieves optimal results when the teacher model is trained for 5 epochs. One possible explanation for this finding is that a teacher model with fewer training epochs aligns better with the initial state of the student model, allowing for a more suitable estimation of sample difficulty.

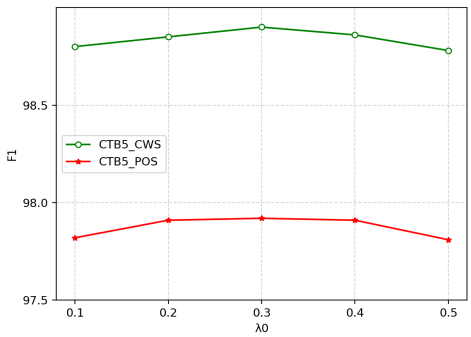


Figure 3: The impact of model-level curriculum learning hyper-parameters λ_0.

Then, we explore the impact of different K values on the BU difficulty metric, which determines the number of dropout times. The experiments are conducted on the CTB6 dataset, and the results are summarized in Table 10. Notably, the model achieves optimal performance when $K = 3$. Therefore, we select $K = 3$ for all the above experiments.

Finally, we evaluate the effect of varying N in the TLC metric. As shown in Table 11, the best performance is achieved when $N = 5$.

E Statistical Significance Test

In this section, we conduct significance testing experiments. Following Wang et al. (2010), we use the bootstrapping method proposed by Zhang et al. (2004), which is operated as follows. In this process, starting with a test set T_0 comprising N test examples, we repeatedly sample N samples from T_0 to form T_1 and then repeat the process for M times to form the test set collection, $\{T_1, T_2, ..., T_M\}$, where M is set to 1000 in our testing procedure. Two systems denoted as A and B, are assessed on the initial test set T_0, resulting in scores a_0 and b_0, respectively. The disparity between the two systems, labeled as δ_0, is calculated as $\delta_0 = a_0 - b_0$. Repeating this process for each test set produces a set of M discrepancy scores, denoted as $\{\delta_0, \delta_1, ..., \delta_M\}$.

Para.	PKU	
	CWS	POS
N=1	98.55	96.49
N=2	98.53	96.49
N=3	98.56	96.52
N=4	98.55	96.51
N=5	**98.71**	**96.64**
N=6	98.52	96.52
N=7	98.56	96.51
N=8	98.55	96.48
N=9	98.53	96.49
N=10	98.55	96.51

Table 11: The impact of N in TLC difficulty metric.

Models		CTB5	
A	B	CWS	POS
BERT+DCL(BU)	BERT	>	>
BERT+DCL(MNLP)	BERT	>	>
BERT+DCL(TLC)	BERT	>	>
BERT+DCL(BU)	BERT+CL(Length)	>	>
BERT+DCL(MNLP)	BERT+CL(Length)	>	>
BERT+DCL(TLC)	BERT+CL(Length)	>	>

Table 12: Statistical significance test of F-score for our method and baselines on the CTB5 dataset.

Following the methodology proposed by Zhang et al. (2004), we compute the 95% confidence interval for the discrepancies (i.e., the 2.5th percentile and the 97.5th percentile) between the two models. If the confidence interval does not overlap with zero, it is affirmed that the differences between systems A and B are statistically significant (Zhang et al., 2004).

Table 12 lists the significant differences between our system and the baseline system, where ">" indicates that the average value of δ exceeds zero, meaning that System A is better than System B; "<" indicates that the average value of δ does not exceed zero, meaning that System A is worse than System B; "~" indicates that there is no significant difference between the two systems. Finally, the comparison also indicates that our models are superior to the baseline.

Accelerating Dense LLMs via L0-regularized Mixture-of-Experts

Zhenyu Zhang [1], Jiudong Yang [2], Zhaowen Tao [1], Meng Chen [3]*
[1] YZW, Chengdu, China
[2] FuTu AI, Shenzhen, China
[3] Wise AI, Melbourne, Australia
zhangzhenyu13@outlook.com, simonyang@futunn.com
taozhaowen@gmail.com, chenmengdx@gmail.com

Abstract

Large language models (LLMs) achieve strong performance but suffer from slow and costly inference. Existing acceleration methods often lead to noticeable performance degradation, while Mixture-of-Experts (MoE) models require extensive computational resources. In this paper, we propose L0-MoE, a lightweight MoE approach using L0-regularization to accelerate dense LLMs nearly without performance loss. Our method introduces a cluster confusion matrix for domain-aware dataset curation and applies dynamic batching for efficient training. Experiments show that L0-MoE achieves up to 2.5x speedup over dense models while maintaining competitive performance, outperforming existing LLM acceleration baselines.

1 Introduction

Large language models (LLMs) have demonstrated remarkable intelligence across various tasks (OpenAI et al., 2024; Gemini-Team et al., 2024; Dubey et al., 2024; Jiang et al., 2023; DeepSeek-AI et al., 2025; Yang et al., 2024), including question answering, mathematics, coding, and content generation. A key insight into their success is the parameter scaling law (Kaplan et al., 2020), which suggests that increasing model size enhances performance across diverse tasks, potentially advancing artificial general intelligence (AGI) (Bubeck et al., 2023). However, larger LLMs incur high inference costs, leading to slower generation speeds and increased computational expenses. Thus, optimizing LLM inference efficiency has become a critical challenge for both academia and industry.

Various approaches have been proposed to accelerate LLM inference, which can be categorized into three main techniques: (1) **Quantization**, including GPTQ (Frantar et al., 2023), SmoothQuant (Xiao et al., 2023), AWQ (Lin et al., 2024b) and DuQuant (Lin et al., 2024a), reduces precision by converting weights and activations from floating-point to lower-bit integer formats, significantly improving efficiency. (2) **Model pruning**, such as LLM-Pruner (Ma et al., 2023) and LLM-Shearing (Xia et al., 2024), removes redundant parameters based on predefined criteria to compress models and accelerate inference. (3) **Knowledge distillation** (Gu et al., 2024; Feng et al., 2024), like reverse-KD (Gu et al., 2024) and Chain-of-Thought (CoT) Distillation (Feng et al., 2024), transfers knowledge from large LLMs to smaller ones using distillation techniques (Hinton et al., 2015), reducing computational demands. While these methods achieve substantial speedup, they often come at the cost of performance degradation, posing challenges for real-world deployment.

Recently, sparsely gated Mixture-of-Experts (MoE) models (Cai et al., 2024), particularly in transformer-based large language models, have significantly improved inference speed optimization. MoE operates on a simple yet effective principle: different model components, known as experts, specialize in distinct tasks or data aspects. For a given input, only relevant experts are activated, reducing computational costs while leveraging a vast pool of specialized knowledge. This scalable and flexible approach aligns with the scaling law, enabling larger model capacities without proportional computational overhead. However, current MoE training focuses on training from scratch or upcycling dense LLMs, both requiring vast computational resources and high-quality corpora. For instance, DeepSeek-V3 (DeepSeek-AI et al., 2024) and Qwen2.5-Max (Yang et al., 2025) were pretrained on 14.8T and 20T tokens, respectively, with additional fine-tuning, making them costly and less accessible. In contrast, *little research has explored leveraging MoE to accelerate inference using a small-scale training corpus (e.g., tens of billions of tokens) while maintaining performance comparable to dense LLMs*. This direction is particularly ap-

*Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 504–513
July 27 - August 1, 2025

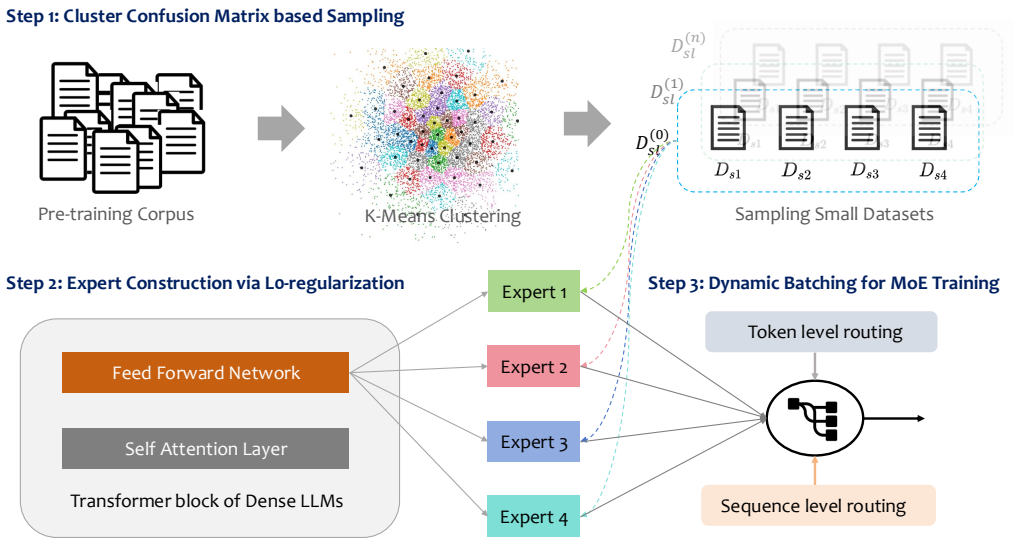


Figure 1: Overview of the L0-MoE Architecture, which includes three main stages: (1) cluster confusion matrix based sampling, (2) expert formation using L0 regularization, and (3) dynamic batching for MoE training. The figure above illustrates the process of building an L0-MoE with four experts over n iterations of dataset sampling.

pealing for large-scale industrial applications with cost-sensitive deployment constraints.

To address this issue, we propose **L0-MoE**, a mixture-of-experts (MoE) model built via L0-regularization (Louizos et al., 2018) using a small, curated **30B**-token corpus. Our approach has two key components: (1) L0-regularization selects critical hidden dimensions in transformer MLPs to form experts. (2) A cluster confusion matrix (CCM)-based sampling method curates the training corpus and schedules dynamic batching. Using the BGE-M3 encoder (Chen et al., 2024) and K-means clustering (Jin and Han, 2010), we extract diverse semantic domains from RedPajama (Weber et al., 2024) to construct expert-relevant sub-datasets. A gating mechanism and dynamic batching optimize training. L0-MoE achieves **2.5×** inference speedup with no obvious performance loss across four benchmarks. Our contributions are as follows: 1) We introduce a novel MoE building method leveraging L0-regularization, enabling efficient LLM inference acceleration with minimal training cost. 2) We propose a CCM-based corpus curation and dynamic batching strategy for effective MoE training. 3) Extensive experiments validate the efficiency of our method in achieving inference speedup while maintaining performance.

2 Preliminary

2.1 L0-regularization

L0-regularization (Louizos et al., 2018) is a powerful technique for feature selection and parameter pruning in neural networks. It imposes a penalty on parameters that deviate from zero, without additional constraints. This approach enhances model efficiency by eliminating unnecessary computations and resources, as irrelevant parameters are pruned and thus not computed. For a given weight matrix $W \in R^{m\times n}$, a mask matrix $Z \in 0, 1^n$ is employed to derive a reduced weight $g(W, Z) \in R^{m\times n0}$, where g selects $n0 < n$ columns from W using Z. Due to the non-differentiable nature of Z, optimizing it is challenging. To address this, the binary hard concrete function is introduced for L0-regularization, as shown in Equation 1.

$$\begin{gathered} u \sim \mathcal{U} \\ s = Sigmoid((log(u) - log(1-u) + loga)/\mathrm{b}) \\ \bar{s} = s(\zeta - \gamma) + \gamma \\ z = min(1, max(0, \bar{s})) \end{gathered} \tag{1}$$

The uniform distribution $\mathcal{U}$ is defined over the interval [0,1]. We set the hyper-parameters as $b = 0.83$, $\zeta = 1.1$, and $\gamma = -0.1$ by following Louizos et al. (2018). Using the learned z, we estimate the proportion of retained weights as $\hat{r} = \frac{sum(z)}{m*n}$. To effectively control the desired retention ratio r for a given weight matrix W, we employ a Lagrangian multiplier (Wang et al., 2019), as described in Equation 2.

$$\mathcal{L}_{l_0} = \lambda_1(\hat{r} - r) + \lambda_2(\hat{r} - r)^2 \tag{2}$$

We initialize the learnable parameters λ_1 and λ_2 to 0 in our experiments. In our approach, r represents the retention ratio of the feed-forward network (FFN) up-projection dimension.

2.2 Mixture of Expert

Mixture of Experts (MoE) (Cai et al., 2024) employs a modular architecture comprising a gating network and multiple expert networks to enhance efficiency and performance through parameter scaling. This architecture partitions the model into several experts, each specializing in specific subsets of input data. MoE utilizes a gating mechanism with a router to dynamically select the appropriate experts for processing incoming inputs, allowing the model to concentrate on relevant features while minimizing unnecessary computations. In our approach, the router is implemented as a linear projection layer $W_{router} \in R^{d \times N}$. MoE incorporates two auxiliary losses (Equation 3), such as the load balancing loss $\mathcal{L}_{balance}$ (Fedus et al., 2022) and the router Z-Loss $\mathcal{L}z$ (Zoph et al., 2022), to promote a balanced distribution of inputs among experts. These losses penalize high values in the logits produced by the gating network, encouraging a more even allocation of tokens to experts.

$$\begin{aligned} \mathcal{L}_{aux} &= \mathcal{L}_{balance} + \lambda \mathcal{L}_z \\ \mathcal{L}_{balance} &= \textstyle\sum_{i=1}^{i=N} (\frac{c_i}{B} - \frac{1}{N})^2 \\ \mathcal{L}_z &= \frac{1}{B} \textstyle\sum_1^B (log(\sum_i^N e^{x_i^{(j)}})) \end{aligned} \tag{3}$$

Here c_i represents the tokens of the ith expert, and N denotes the number of experts. The batch contains B tokens. The logit for the jth token from the ith expert, as determined by the router module, is denoted as $x_i^{(j)}$.

3 Approach

3.1 Cluster Confusion Matrix based Sampling

Given a pretraining corpus, we construct training datasets via the following steps: 1) Randomly sample a small subset without replacement and use the BGE-M3 encoder (Chen et al., 2024) to extract d_{sv}-dimensional semantic vectors for each sample. 2) Apply the K-means clustering algorithm (Jin and Han, 2010) to the semantic vectors to identify K centers $C \in \mathbb{R}^{K \times d_{sv}}$. Divide the small subset into K folds and sample m instances from each fold to form a dataset $D_{sl} = \{D_{s_1}, \ldots, D_{s_K}\}$ for domain semantic learning, where $|D_{s_k}| = m$ for $1 \leq k \leq K$. 3) Repeat steps 1 and 2 for Q iterations to obtain $Q \times K$ centers and Q datasets. For the lth iteration ($l = \{1, 2, \ldots, Q\}$), the cluster centers are $C^{(l)} \in \mathbb{R}^{K \times d_{sv}}$ and the constructed dataset is $D_{sl}^{(l)}$.

We define the clustering confusion matrix (CCM) as per Equation 4, where $\delta = 0.1$ is a hyperparameter, C_i represents the ith center vector, and $CCM[i, l]$ denotes the clustering confusion value for the ith center at iteration l. The hypothesis posits that the semantic domain distance for the ith center between C_i and $C_i^{(l)}$ can be assessed using bidirectional inter-clustering (f_1 and f_2) and intra-clustering (f_3) cosine similarity. A larger semantic domain distance indicates that $D_{sl}^{(l)}$ from the lth iteration divides domains more distinctly. We compute the domain semantic distance using Equation 5 and reorder the Q datasets based on $d_{ds}^{(l)}$. In addition to the initial $D_{sl}^{(0)}$, our datasets now include $Q-1$ ordered datasets $D_{sl}^{ord(l)}$, where $ord(l)$ is the order index.

$$\begin{aligned} CCM[i,l] &= (f_1 + f_2) * f_3 \\ f_1(i,l) &= \tfrac{1}{K} \textstyle\sum_{k=1}^{k=K} e^{1-sim(C_i, C_k^{(l)})} \\ f_2(i,l) &= \tfrac{1}{K} \textstyle\sum_{k=1}^{k=K} e^{1-sim(C_k, C_i^{(l)})} \\ f_3(i,l) &= \delta \tfrac{\sum_{k=1}^{k=K} e^{sim(C_k^{(l)}, C_i^{(l)})}}{\sum_{k=1}^{k=K} e^{sim(C_k, C_i)}} \end{aligned} \tag{4}$$

$$d_{ds}^{(l)} = max(CCM[:,l]) + \frac{\beta}{K} \sum_{i=1}^{i=K} CCM[i,l] \tag{5}$$

3.2 Expert Construction via L0-regularization

We construct the experts using pretrained checkpoints of dense LLMs. The intermediate size of the feed forward network (FFN) layer is d_{int}, and we apply a mask $Z \in R^{d_{int}}$. For each domain subset D_{s_k} ($k \in \{1, 2, \ldots, K\}$) derived from the initial $D_{sl}^{(0)}$, we employ the LLM pretraining loss $\mathcal{L}_{llm}$ along with the L0-regularization loss, as specified in Equation 1, to select $r \in (0, 1)$*100% of the dimensions from d_{int}, following Equation 6.

$$\mathcal{L}_{exp} = \mathcal{L}_{llm} + \mathcal{L}_{l0} \tag{6}$$

To ensure stable training, we gradually adjust r from 100% to the target ratio r^{target}. We freeze all non-MLP parameters of dense LLMs, and the L0-regularization-based training yields K experts, each specialized for distinct semantic domains.

3.3 Dynamic Batching for MoE Training

To train the MoE to effectively select appropriate experts based on inputs, we follow Equation 7, where $\mathcal{L}_{aux}$ is defined in Equation 3. The MoE is initialized with K pre-trained experts and a router for each MoE layer.

$$\mathcal{L} = \mathcal{L}_{llm} + \alpha \mathcal{L}_{aux} \tag{7}$$

We employ a two-loop batch construction strategy during training: 1) domain semantic distance scheduling, where we begin with $D_{sl}^{ord(l)}$ having a

lower d_{ds}; 2) multi-domain gathering scheduling, where samples in $D_{sl}^{ord(l)}$ are arranged in a cyclic sequence order $x_i^1, x_i^2, \ldots, x_i^K$, and we select $p*K$ ($p = \{1, 2, 3, \ldots\}$) samples to form a batch. This scheduling offers two advantages: 1) In the initial iterations, the MoE rapidly learns to select appropriate experts since the domain samples in D_{sl} have been previously encountered by the experts. Consequently, the sequence-level selection capabilities of routers are effectively initialized. 2) As training progresses, the domains in $D_{sl}^{ord(l)}$ gradually transition to different semantic spaces, encouraging routers to select multiple experts for each input sample. This enhances the token-level selection capabilities of the routers.

4 Experiments

4.1 Experimental Setup

Dataset. We train on the RedPajama dataset (Weber et al., 2024), a replicated pre-training corpus for LLaMA models, following prior work (Xia et al., 2024). Evaluation is conducted on four public benchmarks: MMLU (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), HumanEval (Chen et al., 2021), and BigBench Hard (BBH) (Suzgun et al., 2023). Each benchmark evaluates distinct aspects of model performance, offering insights into the strengths and limitations of LLMs.

Baselines. To assess effectiveness and versatility, we evaluate our method on three open-source LLMs: Llama-3-8B (Dubey et al., 2024), Mistral-7B (Jiang et al., 2023), and Qwen2-7B (Yang et al., 2024). Comparisons include L0-regularized MoEs, original LLMs, and inference optimization techniques such as GPTQ quantization (Frantar et al., 2023), LLM Shearing pruning (Xia et al., 2024), and RKD + CoT knowledge distillation (Gu et al., 2024; Feng et al., 2024). For CCM, we run 21 iterations, collecting 30B tokens. Experiments use a cluster/expert size of $K = 64$ with linear warmup, annealing, and a peak learning rate of 1e-4. Further details are in Appendix A.2.

Implementation Details. We train our model using the FSDP framework[1], employing a layer-wise wrapping policy with the Zero-3 parameter sharding strategy, without CPU offloading. For inference during evaluation, we utilize the SGlang framework[2], which is highly optimized for the efficient execution of both dense LLMs and MoEs. All baseline models in our experiments utilize the same SGlang inference framework, ensuring a fair and consistent comparison of inference speeds. Our method is framework-agnostic and can similarly be implemented using other inference frameworks (e.g., vLLM[3]). The primary source of inference acceleration in our work is the proposed L0-regularization-based MoE architecture, not the inference framework itself. To ensure a fair comparison, we strictly adhere to the original evaluation settings for each benchmark. To support future research, we will release our curated dataset and code to enhance the reproducibility of our work.

4.2 Main Results

Table 1 presents the model with the highest performance under our settings. The L0-MoE consistently achieves a 2-2.5x inference speedup across all base LLMs. Additionally, L0-MoE maintains performance comparable to the base LLMs across four benchmarks, with the L0-MoE variant of Mistral even demonstrating a 1% average performance improvement. Table 2 compares these results with other inference acceleration baselines, which, despite achieving some speedup, exhibit noticeable performance degradation.

4.3 Ablation Study

Table 3 presents the ablation study on the CCM module, dynamic batching, and L0-regularization. Removing the K-means clustering from the CCM module results in a performance decline, underscoring the importance of effective sub-dataset curation. For dynamic batching, substituting it with random order or random batch scheduling also leads to degraded performance.

In the context of MoE expert construction, we replace L0-regularization with four alternative methods: 1) **Random MoE** (Zhu et al., 2024): Selects MLP dimensions randomly, serving as a baseline to assess the necessity and effectiveness of dimension selection in expert construction, 2) **Magnitude** (Sun et al., 2023)): Selects the most influential elements in the weight matrix, improving upon traditional magnitude pruning by considering both the weights and their corresponding input activations using the L2 norm, 3) **OBS** (Frantar et al., 2021; Frantar and Alistarh, 2022)): Identifies the most critical dimensions using the OBS Hessian matrix, which encapsulates second-order derivative information of the loss function with respect to model

[1] https://pytorch.org/docs/stable/fsdp.html

[2] https://github.com/sgl-project/sglang

[3] https://github.com/vllm-project/vllm

Model	MMLU	GSM8K	HumanEval	BBH	Average	Speedup
Llama-3-8B	**66.6**	**56.0**	33.5	**57.7**	**53.5**	
Llama-3-8B w/ L0-MoE	66.3	55.9	**33.7**	57.2	53.3	2.0x
Mistral-7B	64.1	52.2	29.3	**56.1**	50.4	
Mistral-7B w/ L0-MoE	**64.8**	**53.6**	**31.1**	55.9	**51.4**	2.1x
Qwen2-7B	70.3	79.9	51.2	**62.6**	66.0	
Qwen2-7B w/ L0-MoE	**70.4**	**80.5**	**52.0**	61.5	**66.1**	**2.5x**

Table 1: Evaluation of different LLMs on MMLU, GSM8K, HumanEval and BBH benchmarks.

Model	MMLU	GSM8K	Speedup
Qwen2-7B	70.3	79.9	-
L0-MoE	**70.4**	**80.5**	2.5x
GPTQ	67.8	73.8	1.8x
LLM Shearing	68.2	75.5	2.6x
RKD + CoT	61.2	60.2	**5.1x**

Table 2: Comparison with other inference acceleration baselines. We employ Qwen2-7B as the base LLM.

Model	MMLU	GSM8K
L0-MoE	**70.4**	**80.5**
CCM w/o K-means	68.2	78.1
w/ random order batching	68.2	75.5
w/ random batch batching	66.6	77.1
Random MoE	48.1	69.6
Magnitude	52.6	69.1
OBS	68.4	74.1
SVD	55.2	73.8

Table 3: Ablation study of CCM and L0-regularization. We conduct experiments on MMLU and GSM8K datasets with Qwen2-7B.

parameters. This approach is crucial for both pruning and quantization, as it helps retain the weights that most significantly impact model performance, 4) **SVD** (Wang et al., 2024)): Decomposes the weight matrix using singular value decomposition and selects the most significant columns. By retaining the largest singular values, it reduces parameter count while preserving essential information. This truncation minimizes compression loss, and layer-wise updates further fine-tune the model to maintain accuracy. The results demonstrate the superiority of L0-MoE over them.

Model	MMLU	#Para.(B)	Speedup
L0-MoE	**70.4**	**23.3**	2.5x
CCM Iter. ($Q = 2$)	68.2	23.3	2.5x
CCM Iter. ($Q = 5$)	68.8	23.3	2.5x
CCM Iter. ($Q = 10$)	69.7	23.3	2.5x
Expert size ($K = 8$)	50.9	4.8	**4.6x**
Expert size ($K = 32$)	69.2	12.7	3.2x

Table 4: Hyper-parameter tuning of sampling iterations (Q) and cluster size (K), keeping 2.8B activated parameters for L0-MoE. Qwen2-7B is the base LLM. #Para.(B) is the number of model parameters.

4.4 Discussion

Our approach involves two critical hyperparameters: the sampling iterations Q in CCM curated datasets and the expert size K in MoE. Table 4 provides a detailed overview of hyperparameter tuning. Increasing the number of iterations for CCM enhances performance but also demands greater computational resources. We find that an initial iteration plus 20 additional iterations suffice to optimize model performance. While increasing the number of experts improves performance, it also reduces inference speed. Therefore, we select an appropriate expert size to balance performance enhancement and LLM acceleration.

Besides, the optimal number of clusters (K) primarily depends on the characteristics of the pre-training corpus; thus, it may not directly transfer to other experiments if the corpus differs significantly. We recommend applying our method to pre-training corpora with abundant topical diversity, such as RedPajama, which contains millions of domains. In such corpora, a larger K can effectively cluster more specialized subsets, enabling CCM to be more effectively applied when constructing L0-MoE models.

To assess model size impact, we conducted further experiments using a 1.5B-parameter Qwen2 model with 64 experts, achieving a 2x speedup without performance degradation. However, due to computational resource constraints, we have not yet experimented with larger models (e.g., 70B parameters). We hypothesize that larger LLMs could potentially achieve even greater speedups. We leave the verification of this hypothesis for larger-scale models as future work.

5 Conclusion and Future Work

In this paper, we propose a novel Mixture-of-Experts based approach to accelerate LLM inference, leveraging clustering confusion matrix for dataset curation, L0-regularization for expert selection, and dynamic batching for efficient training with only 30B tokens. Our method achieves a 2.5× speedup over dense LLMs, outperforming strong baselines nearly without performance loss. Future work will explore scaling our approach to larger LLMs and expanding the corpus size to further enhance L0-MoE performance beyond dense LLMs.

Limitations

We did not compare our method with MoEs such as DeepSeek-MoE (Dai et al., 2024), Qwen-MoE (Qwen-Team, 2024), and Mixtral (Jiang et al., 2024), which scale up model parameters in dense models with immense computational costs, processing trillions of tokens. In contrast, our approach utilizes only 30B tokens, making it more comparable to baseline post-training inference speedup methods.

Despite the promising results, several limitations remain: (1) The dataset for training each expert is selected via sequence-level semantic clustering, introducing exposure bias since MoE expert selection is performed at the token level. (2) The method does not explicitly measure inter-expert differences, potentially leading to redundant parameters that hinder L0-MoE's inference acceleration. Future work should explore token-level dataset partitioning to mitigate exposure bias. Additionally, novel learning paradigms are needed to reduce parameter redundancy and enhance expert routing efficiency.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4.

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. 2024. A survey on mixture of experts. *arXiv preprint arXiv:2407.06204*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, et al. 2024. Deepseek-v3 technical report.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Tao Feng, Yicheng Li, Li Chenglin, Hao Chen, Fei Yu, and Yin Zhang. 2024. Teaching small language models reasoning through counterfactual distillation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5831–5842.

Elias Frantar and Dan Alistarh. 2022. Optimal brain compression: A framework for accurate post-training quantization and pruning. *Advances in Neural Information Processing Systems*, 35:4475–4488.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*.

Elias Frantar, Eldar Kurtic, and Dan Alistarh. 2021. M-fac: Efficient matrix-free approximations of second-order information. *Advances in Neural Information Processing Systems*, 34:14873–14886.

Gemini-Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, et al. 2024. Gemini: A family of highly capable multimodal models.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Xin Jin and Jiawei Han. 2010. *K-Means Clustering*, pages 563–564. Springer US, Boston, MA.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. 2024a. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024b. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100.

Christos Louizos, Max Welling, and Diederik P. Kingma. 2018. Learning sparse neural networks through $l_0 regularization$. In *International Conference on Learning Representations*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. LLM-pruner: On the structural pruning of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, et al. 2024. Gpt-4 technical report.

Qwen-Team. 2024. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters".

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. 2024. Svd-llm: Truncation-aware singular value decomposition for large language model compression. *arXiv preprint arXiv:2403.07378*.

Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2019. Structured pruning of large language models. *arXiv preprint arXiv:1910.04732*.

Maurice Weber, Daniel Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, et al. 2024. Redpajama: an open dataset for training large language models. *arXiv preprint arXiv:2411.12372*.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2024. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *The Twelfth International Conference on Learning Representations*.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report.

Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. 2024. LLaMA-MoE: Building mixture-of-experts from LLaMA with continual pre-training. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15913–15923, Miami, Florida, USA. Association for Computational Linguistics.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*.

A Appendix

This section provides further details on the model architecture, experimental setup (including evaluation tasks, baselines, and hyperparameter settings), and implementation details.

A.1 Model Architecture

Table 5 presents the detailed architecture of the baseline models and L0-MoE. All models incorporate group query attention (GQA) (Ainslie et al., 2023) within the self-attention layer. For the L0-MoE models, the bottom 4 layers (for Qwen2) and 8 layers (for Mistral and Llama3) are configured as dense layers, while the remaining layers are transformed into MoE layers. We select the top-2 experts for each input token.

A.2 Experimental Setups

Evaluation Tasks. We assess performance on four public benchmarks: MMLU (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), HumanEval (Chen et al., 2021), and BigBench Hard (BBH) (Suzgun et al., 2023).

- **MMLU** (Massive Multitask Language Understanding) (Hendrycks et al., 2021) comprises 57 tasks spanning diverse subjects, including STEM (Science, Technology, Engineering, and Mathematics), humanities, social sciences, and specialized domains such as law and ethics.
- **GSM8K** (Grade School Math 8K) (Cobbe et al., 2021) is a benchmark designed to assess the mathematical reasoning capabilities of LLMs, containing 8,500 high-quality elementary math word problems.
- **HumanEval** (Chen et al., 2021) evaluates the code generation capabilities of LLMs through 164 programming tasks, each requiring the model to generate a function that satisfies a given set of test cases.
- **BBH** (BIG-Bench Hard) (Suzgun et al., 2023) is a subset of the larger BIG-Bench dataset, consisting of 23 highly challenging tasks designed to exceed the capabilities of current LLMs. These tasks demand creative problem-solving and deep domain expertise.

Baselines. We compare the L0-regularized MoEs with the original LLMs and other LLM inference optimization methods, including the quantization baseline GPTQ (Frantar et al., 2023), the model pruning baseline LLM Shearing (Xia et al., 2024), and the knowledge distillation baseline RKD + CoT (Gu et al., 2024; Feng et al., 2024).

- **GPTQ** (Frantar et al., 2023) is a block-wise quantization method that extends traditional power-of-two quantization by allowing non-uniform bin widths, enabling a better approximation of the original floating-point value distribution.
- **LLM-Shearing** (Xia et al., 2024) employs structured pruning to construct lightweight, structured LLMs from pretrained checkpoints. It jointly removes attention heads, layers, feed-forward networks (FFNs), and hidden dimensions in an end-to-end manner to optimize efficiency.
- **RKD + CoT**: We apply RKD (Gu et al., 2024) to distill the CoT (Feng et al., 2024) capabilities of Qwen2-7B into Qwen2-1.5B. RKD (Gu et al., 2024) aligns the student model with the teacher's distribution using reverse Kullback-Leibler divergence (KLD), encouraging the student to focus on the most probable outcomes. This helps preserve the quality of the student model's predictions by distilling Chain-of-Thought (CoT) reasoning from the teacher model.

Hyper-parameter Setting. The detailed hyper-parameter settings are presented in Table 7. This includes the hyper-parameters for the clustering confusion matrix (CCM) as well as those for MoE training.

A.3 Comparison with DuQuant

To further validate our method, we compare it with DuQuant (Lin et al., 2024a), a recent quantization technique targeting outlier activations in large language models (LLMs). DuQuant uses rotation and permutation to redistribute outliers, aiming to simplify quantization and improve robustness. We evaluate it on the LLama-3-8B model using MMLU (Hendrycks et al., 2021) and GSM8K (Cobbe et al., 2021) benchmarks. As shown in Table 6, DuQuant suffers noticeable performance degradation, highlighting its limitations. In contrast, our L0-MoE method performs better under the same setting, demonstrating superior accuracy preservation.

Model	Parameters(B)	Layer	Hidden	Q/KV	FFN	MoE FFN	Experts
Llama-3-8B	7.5	32	4096	32/8	14336		
Llama-3-8B w/ L0-MoE	25.1/3.2	32/24	4096	32/8	14336	1024	64:2
Mistral-7B	7.1	32	4096	32/8	14336		
Mistral-7B w/ L0-MoE	24.7/2.9	32/24	4096	32/8	14336	1024	64:2
Qwen2-7B	7	28	3584	28/4	18944		
Qwen2-7B w/ L0-MoE	23.3/2.8	28/24	3584	28/4	18944	1280	64:2

Table 5: Detailed model architecture parameters. We denote the total and activated parameters of the MoEs, as well as the total layers and MoE layers, using the format "32/24", etc. All models utilize GQA, and we present the query/key-value heads. "FFN" refers to the dense decoder MLP size, while "MoE FFN" indicates the intermediate size of the expert for the MoE layer. The total and activated experts are represented as "64:2", etc.

Model	**MMLU**	**GSM8K**	**SpeedUp**
Llama-3-8B	66.6	56.0	-
Llama-3-8B w/ L0-MoE	66.3	55.9	2.0×
DeQuant/W4A4	57.9	51.6	0.46×
DeQuant + LWC/W4A4	62.2	51.1	0.46×

Table 6: Performance comparison of L0-MoE and DuQuant on MMLU and GSM8K benchmarks. Llama-3-8B is the base LLM. Due to the lack of DuQuant support in SGlang, we tested inference speed using naive PyTorch transformers (batch size = 64, sequence length = 1200). Without optimized kernels, DuQuant is slow (0.46x speedup), but future SGlang support could make it comparable to GPTQ (1.8x speedup).

CCM Hyper-parameters	
Q	21
K	64
d_{sv}	1024
$D_{sl}^{(0)}$	12B tokens; $\lvert D_{s_i}^{(0)}\rvert \approx 0.15B$
$D_{sl}^{(l)}, l \geq 1$	0.9B tokens; $\lvert D_{s_i}^{(0)}\rvert \approx 0.007B$
MoE Training Hyper-parameters	
Sequence length	4096
Learning rate	1e-4
Warmup ratio (expert)	0.2
Warmup ratio (MoE)	0.06
Warmup type	Linear
Annealing ratio	0.1
Annealing type	Cosine
Batch tokens	512K
α in Eq. 7	0.01
β in Equation 5	0.5
λ in Eq. 3	0.3
Training epoch	1

Table 7: Hyper-parameters for CCM and MoE training.

Do Multimodal Large Language Models Truly See What We Point At? Investigating Indexical, Iconic, and Symbolic Gesture Comprehension

Noriki Nishida[1], Koji Inoue[2], Hideki Nakayama[3], Mayumi Bono[4], Katsuya Takanashi[5]

[1]RIKEN, [2]Kyoto University, [3]The University of Tokyo,
[4]National Institute of Informatics, [5]The University of Shiga Prefecture
noriki.nishida@riken.jp, bono@nii.ac.jp

Abstract

Understanding hand gestures is essential for human communication, yet it remains unclear how well multimodal large language models (MLLMs) comprehend them. In this paper, we examine MLLMs' ability to interpret indexical gestures, which require external referential grounding, in comparison to iconic gestures, which depict imagery, and symbolic gestures, which are conventionally defined. We hypothesize that MLLMs, lacking real-world referential understanding, will struggle significantly with indexical gestures. To test this, we manually annotated five gesture type labels to 925 gesture instances from the Miraikan SC Corpus and analyzed gesture descriptions generated by state-of-the-art MLLMs, including GPT-4o. Our findings reveal a consistent weakness across models in interpreting indexical gestures, suggesting that MLLMs rely heavily on linguistic priors or commonsense knowledge rather than grounding their interpretations in visual or contextual cues.

1 Introduction

Human communication is inherently multimodal and extends beyond language; nonverbal expressions, particularly hand gestures (hereafter, gestures), are fundamental in conveying meaning and enhancing interaction (McNeill, 1992; Goldin-Meadow, 2003; Kendon, 2004; Kita, 2000). In recent years, multimodal large language models (MLLMs) have gained significant attention across various domains (Yin et al., 2023; Lu et al., 2024; Liu et al., 2023; Sun et al., 2024; Li et al., 2023; Alayrac et al., 2022; Maaz et al., 2024; Su et al., 2023; Zhang et al., 2023). These models excel at integrating textual, auditory, and visual information. However, their ability to accurately interpret gestures, particularly in dynamic real-world communication, remains underexplored.

In this paper, we investigate the extent to which MLLMs can comprehend the meaning and intent behind gestures in real-world communication. We hypothesize that MLLMs, which do not acquire knowledge through direct interaction with their environment, will struggle significantly with ***indexical*** gestures—gestures that rely on external referents. Compared to ***iconic*** gestures (which depict imagery) and ***symbolic*** gestures (which are conventionally defined by cultural norms), indexical gestures require an understanding of external grounding, posing a challenge for MLLMs.

To test this hypothesis, we first constructed a benchmark dataset based on the Miraikan Science Communication (SC) Corpus (Bono et al., 2014; Sakaida et al., 2018), which contains Japanese dialogue transcripts, videos, and gesture descriptions. We defined and manually assigned five gesture type labels (Indexical, Iconic, Symbolic, Mixed, and Others) to 925 gesture instances in this corpus. Figure 1 illustrages examples of indexical, iconic, and symbolic gestures and their corresponding dialogue contexts and human-written descriptions.

Then, using state-of-the-art MLLMs, including GPT-4o (OpenAI, 2024), Gemini 1.5 Pro (Gemini Team, 2024), Qwen2.5-VL (Bai et al., 2025), and LLaVA-NeXT-Video (Li et al., 2024), we generated gesture descriptions based on both video frames and dialogue contexts. These generated descriptions were then evaluated against human-written reference descriptions to assess their validity. Finally, we analyzed performance differences across gesture types to determine whether MLLMs exhibit systematic weaknesses in interpreting certain gesture types.

Our experiments reveal a consistent difficulty across all tested MLLMs in accurately interpreting indexical gestures compared to iconic and symbolic gestures. Further analysis suggests that MLLMs tend to rely on their internal knowledge, derived from text and pretraining, rather than visually recognizing referential grounding of gestures in dynamic environments. These findings indicate that

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 514–524
July 27 - August 1, 2025

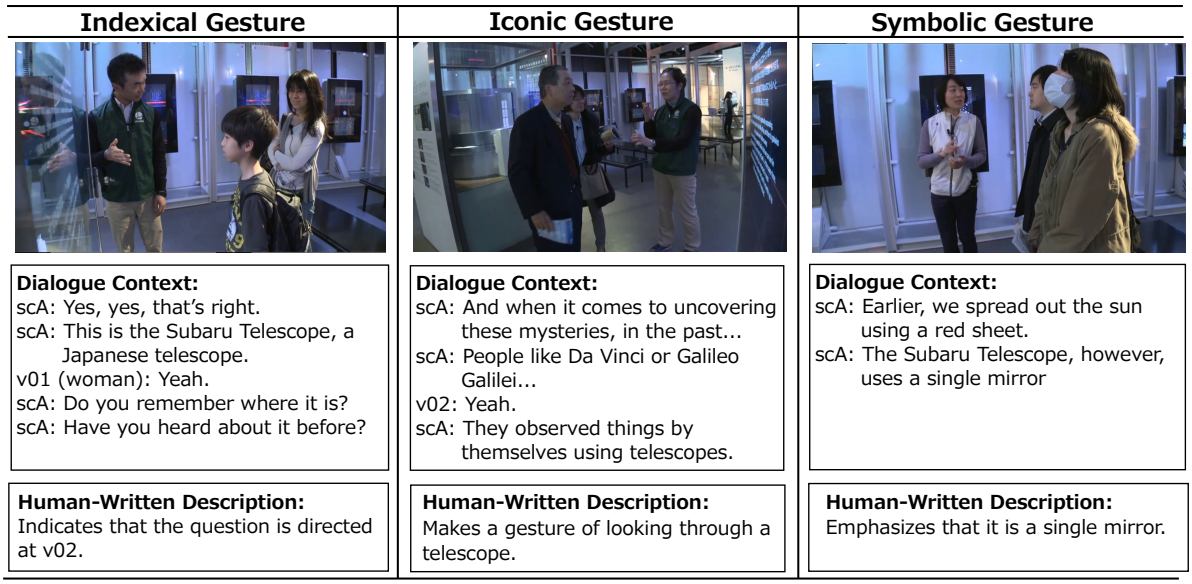


Figure 1: Examples of indexical, iconic, and symbolic gestures, along with their corresponding videos, dialogue contexts, and human-written descriptions. While the original dialogue and descriptions are in Japanese, we provide English translations for clarity.

MLLMs have yet to fully internalize the role of external reference in human communication. Furthermore, unlike most prior work on multimodal language models, which focuses on English inputs and outputs, our study investigates gesture understanding and description generation entirely in Japanese, offering a perspective from a non-English interaction context. We publicly release the annotations along with the source code for data processing and experimentation[1].

2 Dataset Construction

2.1 Building on the Miraikan SC Corpus

We constructed a benchmark dataset by manually annotating *gesture types* to the Miraikan SC Corpus (Bono et al., 2014; Sakaida et al., 2018), a multimodal dataset of video-recorded Japanese conversations between science communicators (SCs) and visitors at the Miraikan science museum in Japan. The corpus contains 35 dialogue sessions, of which 18 sessions include manually-annotated gesture descriptions. Each dialogue session consists of the following data streams synchronized based on timestamps: (1) utterance transcripts, (2) videos captured from 5 fixed cameras, and (3) gesture descriptions. The Miraikan SC Corpus adopts a descriptive approach to gesture description annotation, with a focus on the relevance of gestures to participants' understanding (Bono and Sunaga, 2016). The gesture descriptions are structured into two levels: *interpretation-level descriptions*, which focus on how the gesture intent is interpreted in the communicative context, and *physical-level descriptions*, which detail the physical movements of body parts (face, body, hand, foot)[2]. In our experiments, we used the interpretation-level descriptions of hand movements as references.

2.2 Gesture Type Definitions

We manually defined and assigned one of five gesture types to each of the 925 hand gestures annotated in the Miraikan SC Corpus.

- **Indexical**: Gestures that point to specific referents (e.g., people, objects). Example: Pointing at an exhibit; using hand movements to guide a visitor's gaze.
- **Iconic**: Gestures that visually depict shapes, motions, or spatial configurations of objects or concepts. Example: Drawing the shape of a planet with hands; indicating a mountain's height; mimicking running motions with alternating hand movements.
- **Symbolic**: Gestures defined culturally or socially with conventional meanings. Example: Giving a thumbs-up to indicate "good"; waving to greet

[1] https://github.com/norikinishida/mllm-gesture-eval

[2] For example, "The hand is extended forward and then raised upward" is a physical-level description, while "The gesture indicates the direction of the planet" is an intentional-level description.

Gesture Type	# Examples	Avg. Len. [sec]
Indexical	309 (33.4%)	7.40
Iconic	169 (18.3%)	7.39
Symbolic	8 (0.9%)	6.90
Mixed	20 (2.2%)	9.00
Others	185 (20.0%)	7.30
Uncertain	234 (25.3%)	7.42
Overall	925 (100%)	7.41

Table 1: Dataset Statistics. We show the number of gesture examples for each gesture type. The average duration of each gesture type is also shown.

someone; making a "no" gesture by waving a hand; counting with fingers.

- **Mixed**: Gestures that combine multiple types simultaneously or sequentially. Example: Pointing at an exhibit while drawing a circle around it; pointing at one's eyes while mimicking light entering them.
- **Others**: Gestures outside the above types.

2.3 Annotation Procedure and Statistics

We assigned 3 external annotators[3] to label the same set of 925 gestures with gesture types. All 925 gesture instances were drawn from the 18 sessions in the Miraikan SC Corpus that include manually-annotated gesture descriptions.

We measured inter-annotator agreement across the 3 annotators. Out of the 925 annotated samples, 691 samples (74.7%) had full agreement among all 3 annotators; 220 samples (23.8%) had partial agreement, with 2 annotators assigning the same label and the third assigning a different label; 14 samples (1.5%) had no agreement, with all 3 annotators assigning different labels.

To ensure label reliability, we retained only gesture type labels that were consistently assigned by all annotators. For instances with annotation discrepancies, we assigned a new label, "Uncertain".

Table 1 presents the statistical distribution of the annotated gesture types. Notably, indexical and iconic gestures appear more frequently than symbolic and mixed gestures. This trend aligns with the nature of the Miraikan SC Corpus, which primarily captures exhibit-centered conversations, where pointing and illustrative gestures are commonly used. Furthermore, the average duration of each gesture instance shows no significant variation across indexical, iconic, and symbolic types. This suggests that any observed differences in MLLM-generated gesture descriptions across these types are not due to differences in temporal length, ensuring fair evaluation conditions. Multi-person gestures were extremely rare in the dataset, as the recorded interactions followed structured turn-taking protocols in a science museum setting.

[3] All annotators were native Japanese speakers familiar with the context of the Miraikan SC Corpus, reducing potential cultural mismatches in interpreting symbolic gestures.

3 Experimental Setup

3.1 Multimodal Large Language Models

A variety of multimodal large language model (MLLM) architectures have been proposed in recent years (Yin et al., 2023). While commercial API models such as GPT-4o and Gemini 1.5 Pro have demonstrated superior performance across multiple datasets (Lu et al., 2024; Fu et al., 2024), open-source MLLMs are increasingly being adopted for academic and reproducible research. To ensure broader model coverage, we included the following proprietary models: GPT-4o (OpenAI, 2024), GPT-4o-mini, Gemini 1.5 Pro (Gemini Team, 2024), Gemini 1.5 Flash. In addition, we included the following open-source models: Qwen2.5-VL-7B-Instruct (Bai et al., 2025; Wang et al., 2024; Bai et al., 2023) and LLaVA-NeXT-Video (Liu et al., 2023; Li et al., 2024).

3.2 Gesture Description Generation

To generate gesture descriptions, we provided each MLLM with both dialogue context and video frames leading up to the target gesture. Specifically, the input consisted of video frames and dialogue utterances covering the annotated gesture span, along with the 5-second window of preceding context. A prompt was used to instruct the models to describe the gesture's meaning and intent. Figure 3 in Appendix A illustrates the prompt template used for generation. The same prompt was used across all MLLMs to ensure consistency in evaluation. We focused only on 486 examples labeled as indexical, iconic, or symbolic.

3.3 Evaluation

To evaluate the validity of MLLM-generated gesture descriptions, we used the gesture descriptions from the Miraikan SC Corpus (hand-movement annotations with intentional-level descriptions) as reference descriptions. We employed GPT-4o-mini as the evaluator, prompting it to assess the accuracy and validity of the generated descriptions based on

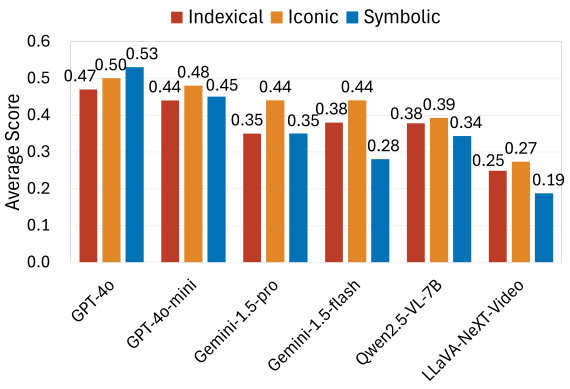


Figure 2: Performance of gesture descriptions generated by MLLMs. Evaluation scores were assigned using GPT-4o-mini, and the average score was computed for each gesture type.

the human-written reference descriptions (Zheng et al., 2023; Chen et al., 2024; Son et al., 2024). The scores range from 0.0 to 1.0, with higher scores indicating greater alignment with the reference descriptions. Figure 6 in Appendix A shows the evaluation prompt. It is worth noting that the human-written reference descriptions occasionally refer to speakers using transcript labels such as "v02" (visitor 02). These speaker labels are included in the dialogue context provided to the models, but are not explicitly linked to any visual entities in the video. This lack of alignment between textual speaker labels and visual grounding may affect the model's ability to resolve referents in some cases.

4 Results and Discussion

4.1 Do MLLMs Struggle with Indexical Gestures?

To evaluate differences in MLLMs' gesture comprehension across types, we averaged the evaluation scores of generated descriptions within each gesture type.

The results (Figure 2) show a clear trend: Indexical gestures consistently received lower scores than iconic gestures across all test models. This pattern held across both proprietary models and open-source models, indicating a consistent weakness in interpreting indexical gestures. Scores for symbolic gestures varied significantly.

These findings confirm our hypothesis that MLLMs struggle with indexical gestures, which require external referential grounding beyond linguistic priors and commonsense knowledge. This limitation underscores a key challenge in current multimodal AI: while MLLMs excel at text-based reasoning, they struggle with context-aware, visually grounded interpretations of referential gestures. Such gestures are essential for human-like communication in dynamic environments, highlighting the need for models that better integrate vision and situational grounding in real-world interactions.

In contrast, iconic gestures, which are often self-contained and can be inferred through linguistic context and commonsense knowledge, are interpreted more reliably. The inconsistency in symbolic gesture scores may suggest that their comprehension is highly model-dependent, influenced by the distribution of cultural conventions in their training data. However, this variability should be interpreted with caution, as the number of symbolic gesture instances is very limited (n=8).

4.2 What Information is Missing for Indexical Gesture Comprehension?

To investigate why MLLMs struggle with indexical gestures and identify the contextual information they may be missing, we conducted additional experiments using GPT-4o. We tested whether augmenting prompts with additional cues would improve the quality of generated descriptions. We explored three modifications: (1) expanding the preceding dialogue window from 5 to 10 seconds for extended dialogue context, (2) incorporating physical-level descriptions of hand movements, and (3) explicitly specifying the gesture type labels. The prompts used for these settings are detailed in Appendix A.

Table 2 presents the results. Extending the dialogue context had minimal effect, suggesting that a longer textual context alone does not significantly improve indexical gesture interpretation. In contrast, providing physical-level descriptions and explicit gesture type labels substantially improved performance, indicating that these gesture-related cues contribute essential information that MLLMs otherwise fail to infer.

These findings suggest that MLLMs' difficulty with indexical gestures is not merely due to insufficient conversational context but rather a lack of understanding of physical motion and referential grounding. While iconic and symbolic gestures can often be self-contained and interpreted using linguistic context and commonsense knowledge, indexical gestures require direct grounding, which MLLMs fail to achieve without external cues.

Additional Cues	Score
No augmentation	0.47
Extended dialogue context	0.48
Physical-level gesture description	0.60
Gesture type label	0.54

Table 2: Impact of additional cues on indexical gesture description generation.

Input Condition	Indexical	Iconic
Full Input	0.47	0.50
Dialogue Only	0.38	0.49
Vision Only	0.43	0.29

Table 3: Evaluation scores for indexical and iconic gestures under different input conditions. “Full Input” uses both video and dialogue, while the other two are ablations.

4.3 Ablation on Modal Contributions

To better understand the respective contributions of visual and linguistic context, we conducted ablation experiments under “dialogue-only” and “vision-only" conditions using GPT-4o. For this analysis, we focused on indexical and iconic gestures.

Table 3 presents the average evaluation scores for each condition. For indexical gestures, removing the visual input led to a noticeable drop in performance (0.47 $\rightarrow$ 0.38), indicating that visual information is essential for resolving referential grounding. In contrast, removing dialogue context had a milder effect (0.47 $\rightarrow$ 0.43), suggesting that indexical interpretation is heavily reliant on visual cues. Iconic gestures, by contrast, were relatively robust to the absence of visual input (0.50 $\rightarrow$ 0.49), but showed a significant drop when dialogue was removed (0.50 $\rightarrow$ 0.29). This suggests that the meaning of iconic gestures is more closely tied to the accompanying linguistic context. For example, a gesture illustrating the height of a mountain is likely to co-occur with utterances about tall mountains, which provides complementary cues that support the interpretation of the gesture's meaning. In contrast, indexical gestures remain difficult to interpret without visual grounding, as dialogue alone often lacks sufficient information to resolve what is being pointed to.

These findings reinforce the hypothesis that indexical gesture comprehension requires visually grounded, referential understanding, while iconic gesture interpretation relies more on context and commonsense associations in language.

4.4 How Are Indexical Gestures Interpreted by MLLMs?

To better understand how MLLMs interpret indexical gestures, we analyzed gesture descriptions generated by GPT-4o alongside their evaluation scores. We found that while GPT-4o often recognized pointing motions as indexical gestures, it frequently misinterpreted their referential intent. For instance, in one case (Figure 6), the human-written description indicated that the pointing gesture serves to disambiguate the (omitted) subject of the utterance, "(...) might have a chance to see through a telescope in the future," without which the subject would remain unclear[4]. However, GPT-4o inferred that the pointing gesture referred to a celestial object on display, likely relying on text-based reasoning rather than external grounding.

These findings suggest that MLLMs prioritize linguistic context and commonsense knowledge over real-world referential resolution. While this strategy suffices for iconic and symbolic gestures, where meaning is largely self-contained, indexical gestures require explicit situational grounding, which MLLMs struggle to achieve.

5 Conclusion

This study investigated MLLMs' ability to comprehend gestures, revealing a consistent weakness in interpreting indexical gestures, which require external referential grounding. Our findings suggest that current models may struggle with interpreting relational gestures such as pointing, possibly due to the absence of dynamic and embodied interactions in their training data. While not conclusive evidence of a fundamental architectural limitation, the results highlight a meaningful gap in multimodal grounding capabilities. A more comprehensive understanding of gesture interpretation may benefit from broader evaluations across diverse conversational contexts, non-English languages, and cultural settings, using datasets that reflect varied interaction styles and referential behaviors. We see our work as a step toward more inclusive and linguistically diverse benchmarks for multimodal understanding.

[4] In Japanese, it is common to omit the subject of a sentence when it can be inferred from context.

Limitations

While this study provides key insights into the limitations of MLLMs in gesture comprehension, several aspects remain to be addressed. (1) Our analysis is based on the Miraikan SC Corpus, which captures interactions in a science museum setting. While this dataset provides rich multimodal information, its domain specificity may limit the generalizability of our findings to other communicative contexts. Future work should explore more diverse datasets, including daily conversations, instructional videos, or multilingual interactions. (2) While our prompts were designed to elicit natural gesture interpretations from MLLMs, they did not fully replicate the annotation guidelines used for the original human-written descriptions in the Miraikan SC Corpus. This gap in prompting design may have contributed to certain discrepancies in interpretation. (3) We employed LLM-based evaluation (GPT-4o-mini) to assess the quality of gesture descriptions. While LLM-based evaluation has been shown to be reliable in many tasks, it remains a proxy measure and may not fully capture the nuances of human interpretation of gestures. We acknowledge this limitation and recommend that future work incorporate human evaluations using consistent rubrics to validate automated assessment more robustly. (4) Symbolic gestures, in particular, are culturally dependent, and the performance variability across models suggests that training data composition plays a major role. Expanding evaluations to different language models and cultural contexts would help clarify whether MLLMs truly internalize gesture meaning or simply reflect training biases.

Acknowledgments

We would like to thank the anonymous reviewers for their thoughtful and insightful comments, which we found very helpful in improving the paper. This work was supported by JSPS KAKENHI Grant Numbers JP22B102, JP22H05013, JP22H05015, and JP21K17815.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millicah, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a visual language model for few-shot learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Mayumi Bono, Hiroaki Ogata, Katsuya Takanashi, and Ayami Joh. 2014. The practice of showing 'who i am': A multimodal analysis of encounters between science communicator and visitors at science museum. In *Universal Access in Human-Computer Interaction. Universal Access to Information and Knowledge*, pages 650–661, Cham. Springer International Publishing.

Mayumi Bono and Masashi Sunaga. 2016. A proposal of annotation scheme for body movement based on participants' understandings. In *JSAI SIGs Conference Papers, SIGSLUD-B503*.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yinuo Liu, Yaochen Wang, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. 2024. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. *arXiv preprint arXiv:2402.04788*.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Rongrong Ji, and Xing Sun. 2024. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *Preprint*, arXiv:2405.21075.

Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Susan Goldin-Meadow. 2003. *Hearing Gesture: How Our Hands Help Us Think*. Harvard University Press, Cambridge, MA.

Adam Kendon. 2004. *Gesture: Visible Action as Utterance*. Cambridge University Press, Cambridge.

Sotaro Kita. 2000. How representational gestures help speaking. In David McNeill, editor, *Language and Gesture*, pages 162–185. Cambridge University Press, Cambridge, UK.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc.

Chaochao Lu, Chen Qian, Guodong Zheng, Hongxing Fan, Hongzhi Gao, Jie Zhang, Jing Shao, Jingyi Deng, Jinlan Fu, Kexin Huang, et al. 2024. From gpt-4 to gemini and beyond: Assessing the landscape of mllms on generalizability, trustworthiness and causality through four modalities. *arXiv preprint arXiv:2401.15071*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. 2024. Video-ChatGPT: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12585–12602, Bangkok, Thailand. Association for Computational Linguistics.

David McNeill. 1992. *Hand and Mind: What Gestures Reveal About Thought*. University of Chicago Press.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Rui Sakaida, Ryosaku Makino, and Mayumi Bono. 2018. Preliminary analysis of embodied interactions between science communicators and visitors based on a multimodal corpus of Japanese conversations in a science museum. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Guijin Son, Hyunwoo Ko, Hoyoung Lee, Yewon Kim, and Seunghyeok Hong. 2024. Llm-as-a-judge & reward model: What they can and cannot do. *arXiv preprint arXiv:2409.11239*.

Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. 2023. PandaGPT: One model to instruction-follow them all. In *Proceedings of the 1st Workshop on Taming Large Language Models: Controllability in the era of Interactive Assistants!*, pages 11–23, Prague, Czech Republic. Association for Computational Linguistics.

Guangzhi Sun, Wenyi Yu, Changli Tang, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, Yuxuan Wang, and Chao Zhang. 2024. video-salmonn: speech-enhanced audio-visual large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*.

Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 543–553, Singapore. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Prompts

This appendix presents the prompts used for gesture description generation and evaluation in our experiments.

A.1 Gesture Description Generation Prompts

To assess MLLMs' ability to interpret gestures, we designed a baseline prompt and explored modifications incorporating external cues to improve Indexical gesture comprehension. Figure 3 presents the baseline prompt, where MLLMs generate gesture descriptions using only dialogue context and video frames without additional guidance. Figure 4 shows the prompt with physical-level descriptions, providing explicit details on hand movements to help MLLMs recognize gestures based on motion rather than relying solely on textual context. Figure 5 illustrates the prompt with explicit gesture

type labels (Indexical, Iconic, or Symbolic), guiding MLLMs to distinguish gestures based on their communicative function.

A.2 Evaluation Prompt

To evaluate the quality of generated gesture descriptions, we employed GPT-4o-mini as the evaluator. Figure 6 presents the evaluation prompt, where GPT-4o-mini was provided with both the generated description and the human-written reference description and instructed to assess their accuracy and validity on a 0.0 to 1.0 scale.

B Annotation Policy and Labeling Considerations

We used the "Mixed" category to account for cases where gestures could not be reliably separated into distinct types due to overlapping characteristics. While a multi-label annotation scheme (e.g., tagging both "indexical" and "iconic") may offer a finer-grained representation, we prioritized high inter-annotator agreement and interpretability in this initial study. We consider multi-label annotation a promising direction for future work.

C Effect of Translation into English

To examine the extent to which language affects MLLM performance, we conducted an additional experiment in which all inputs were translated into English. Specifically, for each gesture instance, we translated the Japanese dialogue context, the reference description, and the instruction prompts (for description generation and evaluation) into English using GPT-4o-mini, We then prompted GPT-4o to generate gesture descriptions in English, using the same prompt structure as in the Japanese setting. Evaluation was performed using GPT-4o-mini, which assessed the generated English descriptions against the translated reference descriptions, using the same rubric as described in Section 3.3.

The results are summarized in Table 4. Interestingly, GPT-4o's performance on indexical gestures improved in the English translation setting (0.47 → 0.56). One possible explanation is that Japanese often omits subjects and relies heavily on context, making utterances and referents more ambiguous. When translated into English, these omissions are often filled in with explicit pronouns like "you" or "he," reducing referential ambiguity. In contrast, the score for symbolic gestures declined in the English setting (0.53 → 0.44), likely due to cultural dependency and meanings being lost in translation. Symbolic gestures often presuppose culturally shared meanings that are implicitly understood within the original culture (e.g., placing hands together to express gratitude or prayer). Such associations are not always explicitly represented in English training corpora, making it difficult for models to infer the intended meaning when gestures are translated out of their cultural context. Note that symbolic gestures are underrepresented in our dataset (n = 8), making this category more susceptible to variance; the observed decline should therefore be interpreted with caution.

Language	Indexical	Iconic	Symbolic
Japanese	0.47	0.50	0.53
English	0.56	0.52	0.44

Table 4: Comparison of gesture description accuracy in Japanese vs. translated English setting (GPT-4o).

以下の会話文と対話シーンの画像系列に基づいて、シーンの終盤で行われたジェスチャーの意味を説明してください。[*Based on the following conversation and a series of images depicting the dialogue scene, explain the meaning of the gesture performed towards the end of the scene*]

会話文：[*Conversation* :]
scA: きょ，星がキラキラ集まったのが，もっと実は，もっともっと外にあります．[*The stars are actually gathered together in a sparkling way, and there are even more, much farther away.*]
v01: もっと(す)．[*More.*]
scA: そうそう，もっと [*Yes, yes, more*]
v04: へーえーえ．[*Oh.*]
v01: もっとあるだって，りゅうせい．[*There are even more meteors.*]
scA: もっと，そう，もっといっぱいあるんです．[*More, yes, there are many more.*]
v03: (あんどろめだー;アンドロメダ)(のね) [*Andromeda.*]
scA: あのね，望遠鏡将来見る機会あるかもしれません．[*Well, in the future, (you) might have a chance to see them through a telescope.*]
scA: あの [*You know...*]

ジェスチャーの説明: [*Gesture Explanation* :]
このシーンの終盤で行われたジェスチャーは、指を使って何かを指し示す動作です。このジェスチャーは、会話の内容に関連して、特定の場所や物体を示すために行われたものと考えられます。具体的には、星や天体に関する話題が出ているため、展示物やディスプレイ上の天体を指し示して、説明を補足している可能性があります。指差しは、視覚的な情報を強調し、聞き手に具体的なイメージを持たせるための効果的な手段です。[*The gesture performed toward the end of this scene involves using a finger to point at something. This gesture is likely made to indicate a specific location or object in relation to the conversation. Specifically, given that the topic revolves around stars and celestial bodies, the gesture may be used to point at an exhibit or a celestial object displayed on a screen to supplement the explanation. Pointing gestures serve as an effective means to emphasize visual information and help the listener form a clearer mental image.*]

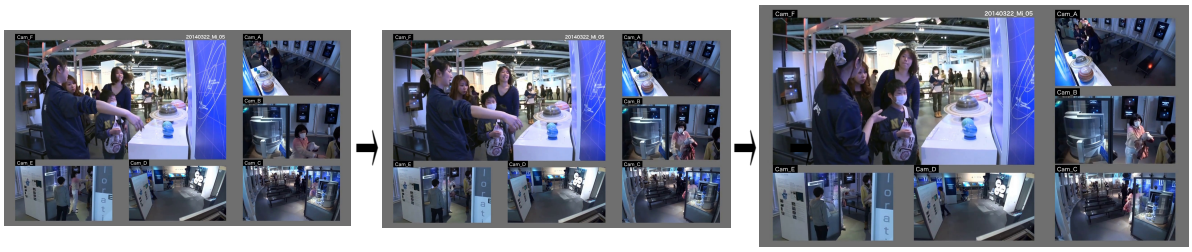


Figure 3: Example prompt used for gesture description generation. Red text indicates variables that change per input instance, while blue text represents the actual output generated by GPT-4o. Each line has been translated into English, with the translation appended in brackets ('[', ']') at the end of each line.

以下の会話文と対話シーンの画像系列に基づいて、シーンの終盤で行われたジェスチャーの意味を説明してください。[*Based on the following conversation and a series of images depicting the dialogue scene, explain the meaning of the gesture performed towards the end of the scene.*]

会話文：[*Conversation* :]
{utterances}

ジェスチャーの物理的な観点からの説明: [*Explanation from the physical perspective of the gesture:*]
{physical_level_description}

ジェスチャーの説明: [*Gesture Explanation* :]

Figure 4: Example prompt used for gesture description generation with physical-level descriptions as additional context. The physical descriptions provide details on the hand movements associated with the gesture, aiming to enhance MLLMs' ability to interpret indexical gestures by incorporating motion-related cues.

以下の会話文と対話シーンの画像系列に基づいて、シーンの終盤で行われたジェスチャーの意味を説明してください。[*Based on the following conversation and a series of images depicting the dialogue scene, explain the meaning of the gesture performed towards the end of the scene.*]

会話文: [*Conversation*:]
{utterances}

ジェスチャータイプ: [*Gesture Type:*]
{gesture_type_label}

ジェスチャーの説明: [*Gesture Explanation*:]

Figure 5: Example prompt used for gesture description generation with explicit gesture type labels as additional context. The gesture type label (Indexical, Iconic, or Symbolic) is provided to guide MLLMs in interpreting the gesture's communicative function more accurately.

以下は、あるジェスチャーの説明文のペアです。1つはモデルによって生成された説明文であり、もう1つは人手で作成された正解説明文です。生成された説明文が正解説明文の意味をどの程度含んでいるかを評価してください。評価は以下の基準に基づき、0.00から1.00の間のスコアで示してください。また、その理由を簡潔に説明してください。[*The following is a pair of descriptions for a gesture. One is a description generated by a model, and the other is a ground truth description created manually. Evaluate the extent to which the generated description includes the meaning of the ground truth description. The evaluation should be given as a score between 0.00 and 1.00 based on the following criteria, along with a brief explanation of the reasoning.*]

評価基準: [*Evaluation Criteria:*]
- 1.00: 生成された説明文が正解説明文の意味を完全に含んでいる。[*The generated description fully includes the meaning of the ground truth description.*]
- 0.75: 生成された説明文が正解説明文の意味の大部分を含んでいるが、いくつかの細部が欠けている。[*The generated description includes most of the meaning of the ground truth description but lacks some details.*]
- 0.50: 部分的に一致しているが、重要な情報が欠けている。[*The descriptions partially match, but important information is missing.*]
- 0.25: わずかな関連性があるが、大部分が一致していない。[*There is slight relevance, but most of the meaning does not match.*]
- 0.00: 全く意味が一致していない。[*The meanings do not match at all.*]

ジェスチャーのタイプ: 指標的 [*Gesture Type: Indexical*]

生成された説明文: [*Generated Description:*]
このシーンの終盤で行われたジェスチャーは、指を使って何かを指し示す動作です。このジェスチャーは、会話の内容に関連して、特定の場所や物体を示すために行われたものと考えられます。具体的には、星や天体に関する話題が出ているため、展示物やディスプレイ上の天体を指し示して、説明を補足している可能性があります。指差しは、視覚的な情報を強調し、聞き手に具体的なイメージを持たせるための効果的な手段です。[*The gesture performed toward the end of this scene involves using a finger to point at something. This gesture is likely made to indicate a specific location or object in relation to the conversation. Specifically, given that the topic revolves around stars and celestial bodies, the gesture may be used to point at an exhibit or a celestial object displayed on a screen to supplement the explanation. Pointing gestures serve as an effective means to emphasize visual information and help the listener form a clearer mental image.*]

正解説明文: [*Ground Truth Description:*]
「望遠鏡将来見る機会があるかもしれません」の発話の主語がvであることを示す [*The gesture indicates that the (omitted) subject of the utterance "(...) might have a chance to see through a telescope in the future" is v.*]

評価スコア: 0.25 [*Evaluation Score: 0.25*]

理由: 生成された説明文は、指を使って何かを指し示す動作についての説明を行っていますが、正解説明文の具体的な内容（「望遠鏡将来見る機会があるかもしれません」の発話の主語がvであることを示す）とは関連性が薄いです。生成された説明文は、ジェスチャーの一般的な機能や目的について述べていますが、正解説明文の重要な情報（発話の主語や具体的な文脈）を含んでいないため、わずかな関連性があるものの、大部分が一致していないと評価しました。[*Explanation:The generated description explains a pointing gesture but is only loosely related to the meaning of the ground truth description. While the generated description discusses the general function and purpose of pointing, it does not include the key information from the ground truth description—specifically, that the gesture is indicating the subject of the utterance. As a result, while there is slight relevance, most of the meaning does not match, leading to a score of 0.25.*]

Figure 6: Example prompt used for gesture description evaluation. Red text indicates variables that change per input instance, while blue text represents the actual output generated by GPT-4o. Each line has been translated into English, with the translation appended in brackets ('[', ']') at the end of each line. Note: In the original Japanese utterance "(...) might have a chance to see through a telescope in the future," the subject is omitted, which is a common phenomenon in Japanese when the subject can be inferred from context. In this case, the pointing gesture serves to disambiguate the subject of the utterance, without which the referent would remain unclear. In our dataset, "v" denotes a visitor in the dialogue. Each participant is labeled in the transcript, with identifiers such as "sc" for the science communicator and "v01", "v02", etc., for individual visitors.

Fast or Slow? Integrating Fast Intuition and Deliberate Thinking for Enhancing Visual Question Answering

Songtao Jiang[1*], Chenyi Zhou[1*] Yan Zhang[2], Yeying Jin[3], Zuozhu Liu[1,4†]

[1]Zhejiang University, [2]Byte Dance, [3]National University of Singapore
[4]Zhejiang Key Laboratory of Medical Imaging Artificial Intelligence
Correspondence[†]: zuozhuliu@intl.zju.edu.cn

Abstract

Multimodal large language models (MLLMs) still struggle with complex reasoning tasks in Visual Question Answering (VQA). While current methods have advanced by incorporating visual prompts, our study uncovers critical limitations: these approaches indiscriminately annotate all detected objects for every visual question, generating excessive visual markers that degrade task performance. This issue stems primarily from a lack of focus on key visual elements, raising two important questions: *Are all objects equally important, and do all questions require visual prompts?* Motivated by Dual Process Theory, which distinguishes between instinctive and deliberate cognitive modes in human reasoning, we propose FOCUS, a plug-and-play approach that dynamically adapts to the complexity of questions, combining fast intuitive judgments with deliberate analytical reasoning to enhance the vision-language reasoning capability of the MLLM. For straightforward questions, FOCUS supports efficient zero-shot reasoning. For more complex tasks, it employs the conceptualizing before observation strategy to highlight critical elements. Extensive experiments on four benchmarks—ScienceQA, TextQA, VizWiz, and MME—demonstrate that FOCUS consistently improves the performance of both open-source and black-box MLLMs, achieving significant gains across all datasets. Ablation studies further validate the importance of combining diverse cognitive strategies with refined visual information for superior performance. Code will be released.

1 Introduction

Multimodal large language models (MLLMs) have demonstrated promising capabilities in visual question answering (VQA) tasks (OpenAI, 2023b; Liu et al., 2023b; Chen et al., 2023). However, MLLMs still face challenges in addressing complex scenarios, particularly those requiring fine-grained visual perception and the ability to effectively leverage visual information for reasoning tasks (Chen et al., 2024; Jiang et al., 2024b).

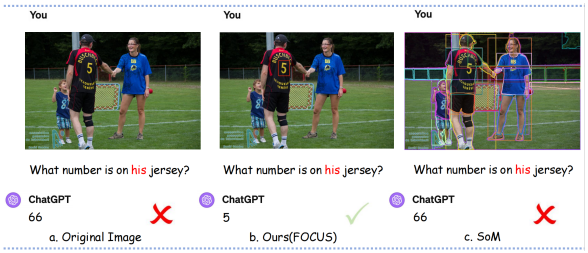


Figure 1: Comparison of methods: baseline (original image), FOCUS, and SoM (Yang et al., 2023) (previous SoTA method). See Appendix A.1 for more discussion.

Current approaches (Yang et al., 2023; Cai et al., 2024) employ segmentation models to generate visual prompts for all questions by annotating all objects in VQA images as visual markers, aiming to enhance MLLMs' attention to visual information for improving accuracy. While effective, as shown in Figure 1, this coarse-grained guidance of adding visual markers to all objects fails to emphasize key elements. Consequently, MLLMs' attention is often distracted by redundant information, while also introducing significant computational overhead.

In this paper, we introduce FOCUS, a novel approach inspired by *Dual Process Theory* (Kahneman, 2011), which delineates two cognitive modes: a fast, instinctive mode (System 1) and a slower, deliberate mode (System 2). FOCUS enables MLLMs to dynamically alternate between rapid intuitive judgments and thorough analytical reasoning, adapting seamlessly to questions of varying complexity. Specifically, for each visual question, FOCUS first assesses its difficulty level. For simple questions, the MLLM performs efficient zero-shot reasoning. For more complex questions, FOCUS employs a *conceptualizing before observation* strat-

* Equal contribution.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 525–534
July 27 - August 1, 2025

egy, which identifies and highlights critical visual elements using a segmentation model, enabling the MLLM to reason based on refined visual inputs. Prior research has demonstrated that MLLMs struggle when visual elements in the input are ambiguous. As illustrated in Figure 1, FOCUS addresses this limitation by guiding MLLMs to focus on key visual elements, thereby unlocking their full reasoning potential. Furthermore, FOCUS not only enhances the accuracy of responses to complex questions but also optimizes computational efficiency, improving the overall performance.

We applied FOCUS to five MLLMs with diverse architectures and parameter scales, including three open-source models: LLaVA-1.5 (Liu et al., 2023a), MiniGPT4-V2 (Chen et al., 2023), and InstructBLIP (Dai et al., 2023), as well as two black-box models: GPT-4V (OpenAI, 2023b) and Gemini Pro (Gemini Team, Google, 2023). We evaluated their performance on four popular benchmarks, demonstrating that FOCUS consistently enhances model capabilities. Notably, FOCUS combined with LLaVA-1.5-13B achieves state-of-the-art (SoTA) performance across all four benchmarks, while FOCUS with LLaVA-1.5-7B delivers performance comparable to LLaVA-1.5-13B. Compared to the previous SoTA method, our approach not only surpasses SoM on all datasets but also reduces inference time by nearly 44%. Additionally, extensive analysis and ablation studies validate the effectiveness of our dual-thinking strategy and demonstrate its capability to achieve attention calibration.

2 Methods

2.1 Problem Definition

In this paper, we denote the dataset as $\mathcal{D} = \{(I_m, Q_m, A_m)\}_{m=1}^{M}$ where M is the number of data samples. The goal of our tasks is to develop a mapping function $\mathcal{F}(\cdot)$ that can generate answers in response to the questions, represented as:

$$A = \mathcal{F}(I, Q), \tag{1}$$

Here, I denotes the visual input. Q represents the natural language question pertaining to the visual input I. A is the output of the MLLM $\mathcal{F}(\cdot)$. FOCUS begins by evaluating question complexity to determine whether to apply the conceptualizing before observation strategy in the inference process. The overview of FOCUS is demonstrated in Figure 2. The following sections will delve into these components in detail.

2.2 Question Complexity Evaluation

Past research (Xiong et al., 2023) has shown that large language models effectively express their confidence in answers by mimicking human behavior, though they are prone to overconfidence. Building on this insight, we guide the MLLM to indicate their confidence before answering the question. If the MLLM demonstrates high confidence, we classify the question as simple. Otherwise, it is considered complex. Additionally, to mitigate the issue of overconfidence, we employ the self-consistency (Wang et al., 2022) method along with strict confidence evaluation criteria.

Specifically, we first set a high temperature for the MLLM and utilized the prompt strategy from VizWiz (Gurari et al., 2018) to activate the MLLM, generating N responses $\{r_i\}_{i=1}^{N}$, where r_i is either "Answerable" or "Unanswerable". A question is deemed simple if all responses are "Answerable." Otherwise, it is considered complex. In this paper, we set N to three. We also discuss the effectiveness of the selected prompt and chosen N in Appendix B.1.

2.3 Conceptualizing before Observation

This strategy is specifically designed to address the complex question where the original visual information is confusing or insufficient. By conditioning the segmentation process on key extracted elements, it ensures that MLLMs can concentrate on the most crucial parts of the image corresponding to the question. To be specific, the first step involves extracting K key elements $\{k_i\}_{i=1}^{K}$ from the question Q using the language model. Then the key elements $\{k_i\}_{i=1}^{K}$ are combined with bounding boxes generated by the segmentation model $\mathcal{S}$, which serves as an open-set object detector. The segmented regions $\{s_i\}_{i=1}^{K}$ are defined as:

$$s_i = \mathcal{S}(I, k_i) \tag{2}$$

The final processed image I' is produced by aggregating the relevant segments:

$$I' = \sum_{i \in K} s_i, \tag{3}$$

Finally, the processed image I', along with the original question Q, is then fed into the MLLM to generate a more accurate response. In this paper, we used GPT-3.5-turbo (OpenAI, 2023a) as the language model and Grounded-SAM (Ren

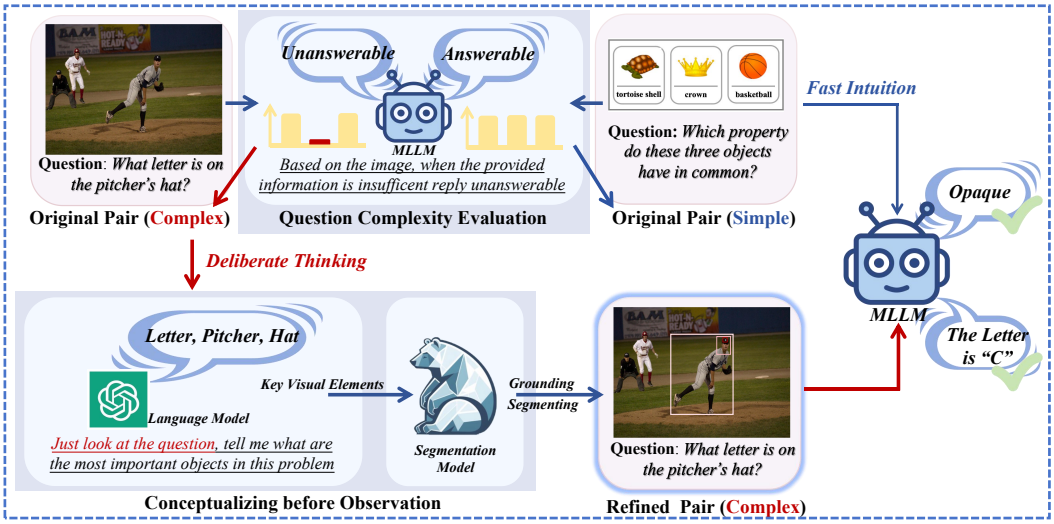


Figure 2: Overview of our model pipeline. (1) Question complexity evaluation for MLLM-based behavior to determine which questions require deliberate thinking. (2) Conceptualizing before observation, helping the model identify the most important visual information in the image.

et al., 2024) as the segmentation model. The effectiveness of Grounded-SAM is discussed in Appendix B.2, while the effectiveness of GPT-3.5-turbo is discussed in Appendix B.3.

3 Experiement

Datasets. We evaluate FOCUS using three popular open-source MLLMs and two black-box MLLMs. The evaluation is conducted on four datasets specifically chosen to assess distinct capabilities: ScienceQA (Lu et al., 2022), TextVQA (Singh et al., 2019), VizWiz (Gurari et al., 2018), and MME (Fu et al., 2023). For ScienceQA, TextVQA, and VizWiz, we measure accuracy, while for MME, we utilize the total score to evaluate overall performance. These datasets collectively address challenges in *logical reasoning, text recognition, real-world visual understanding, and generalization*, offering a comprehensive assessment of FOCUS's capabilities. Additional details are in Appendix B.4.

Baselines. We evaluate two types of baselines for comparison. The first category comprises training-based approaches (Zhang et al., 2023; Li et al., 2023), which leverage additional data or larger models. The second category includes inference-based methods, which utilize techniques such as cot reasoning or visual prompts (Yang et al., 2023; Zheng et al., 2023; Mitra et al., 2024).

3.1 Main Results

Comparison with Previous Methods: As shown in Table 1, our method significantly outperforms previous approaches. Compared to the best-performing method, SoM, our approach not only achieves better performance but also offers a cost advantage, as it does not require adding unnecessary visual markers to the entire image.

Applicability to Different MLLMs: We evaluated FOCUS on multiple MLLMs, as detailed in Table 2 and Table 3. The results show that FOCUS consistently improves performance across diverse model architectures and parameter scales. For black-box MLLMs, we tested on a randomly selected subset of 250 samples from ScienceQA due to the cost.

3.2 Ablation Study and Analysis

Ablation of Thinking Strategies. As shown in Table 4, we assess the impact of Fast Intuition and Deliberate Thinking in FOCUS. Results reveal that uniformly applying a single strategy across all questions reduces performance. While Deliberate Thinking offers limited gains, it incurs significant computational costs. Conversely, relying solely on Fast Intuition hampers the MLLM's ability to address complex questions. Thus, adapting strategies based on question complexity is essential for optimal performance.

How does Deliberate Thinking outperform SoM? We compare the superiority of our Deliberate Thinking approach, which emphasizes Conceptualizing before Observation, with SoM's coarse-grained strategy of globally adding visual markers. As shown in Table 5, our Deliberate Thinking method consistently outperforms SoM across all

Method	LLM	ScienceQA(%)	TextVQA(%)	VizWiz (%)	MME
BLIP-2 (Li et al., 2023)	Vicuna-13B	61.0	42.5	19.6	1293.8
IDEFICS-9B (Laurençon et al., 2024)	LLaMA-7B	—	25.9	35.5	—
IDEFICS-80B (Laurençon et al., 2024)	LLaMA-65B	—	30.9	36.0	—
Qwen-VL-Chat (Bai et al., 2023)	Qwen-7B	68.2	61.5	38.9	1487.5
LVIS-INSTRUCT4V (Wang et al., 2023)	Vicuna-13B	69.0	62.1	51.4	1572.0
CCoT (Mitra et al., 2024)	Vicuna-7B	68.7	58.9	51.1	—
DDCoT (Zheng et al., 2023)	Vicuna-7B	67.6	57.9	50.5	—
DCoT (Jia et al., 2024)	Vicuna-7B	69.3	59.1	51.9	—
SoM (Yang et al., 2023)	Vicuna-13B	71.3	61.5	54.0	1540.1
FOCUS + LlaVA-1.5-7B	Vicuna-7B	70.2	60.1	54.5	1528.4
FOCUS + LlaVA-1.5-13B	Vicuna-13B	**74.4** (+3.1)	**63.6** (+2.1)	**58.5** (+4.5)	**1551.0** (+10.9)

Table 1: Comparison of methods across ScienceQA, TextVQA, VizWiz, and MME. Rows highlighted in yellow represent training-based methods, green represent inference-based methods, which are our primary comparison targets, and pink highlight FOCUS.

Method	LLM	ScienceQA(%)	TextVQA(%)	VizWiz (%)	MME
LLaVA-1.5	Vicuna-7B	66.8	58.2	50.0	1510.7
LLaVA-1.5	Vicuna-13B	71.6	61.3	53.6	1531.3
LLaVA-1.5 + FOCUS	Vicuna-7B	**70.2** (+3.4)	**60.1** (+1.9)	**54.5** (+4.5)	**1528.4** (+17.7)
LLaVA-1.5 + FOCUS	Vicuna-13B	**74.4** (+2.8)	**63.6** (+2.3)	**58.5** (+4.9)	**1551.0** (+19.7)
MiniGPT4-V2	LLaMA-7B	56.7	34.1	44.1	1316.5
MiniGPT4-V2 + FOCUS	LLaMA-7B	**59.6** (+2.9)	**36.2** (+2.1)	**47.4** (+3.3)	**1334.3** (+17.8)
InstructBLIP	Vicuna-7B	60.5	50.1	34.5	1174.5
InstructBLIP + FOCUS	Vicuna-7B	**62.3** (+1.8)	**52.3** (+2.2)	**36.8** (+2.3)	**1189.8** (+15.3)

Table 2: Comparative performance of open-source MLLMs across ScienceQA, TextVQA, VizWiz and MME. Both LLaVA-1.5 and MiniGPT-4-V2 use MLP as the connecting component between the ViT (Dosovitskiy, 2020) and LLM backbones, whereas InstructBLIP employs Q-Former (Li et al., 2023)

datasets, demonstrating its effectness in VQA.

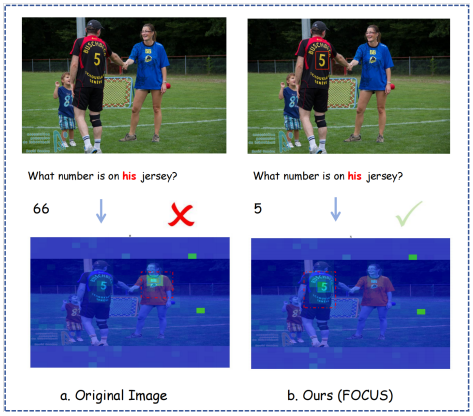


Figure 3: Visual attention visualization in LLaVA-1.5. The green areas indicate higher attention weights.

Visual Attention Calibration. We assessed whether FOCUS's fine-grained visual markers enhance MLLMs' ability to filter distractions by analyzing the attention patterns of LLaVA-1.5. As shown in Figure 3, in the original image, the MLLM mistakenly focuses on the woman's jersey number, resulting in an incorrect answer. In contrast, FOCUS guides the MLLM's attention to the relevant image regions. This highlights FOCUS's ability to help MLLMs identify critical visual information, enhancing their perceptual accuracy.

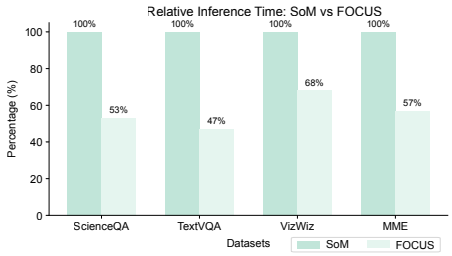


Figure 4: Relative inference time comparasion

Cost Analysis. As illustrated in Figure 4, our method achieves faster inference while maintaining superior performance compared to SoM. This efficiency stems from our approach's ability to skip the unnecessary addition of visual prompts for certain questions, as well as its avoidance of annotating all objects in the image—a process that is both time-consuming and detrimental to performance. These advantages demonstrate that our method is a

cost-effective solution for VQA tasks.

Analysis of Answerable and Unanswerable Proportions. We perform an analysis of the proportions of answerable versus unanswerable responses across different VLMs and benchmarks. Specifically, we compute average unanswerable proportions for three base VLMs, obtaining 39.00% for LLaVA-1.6, 42.40% for LLaVA-1.5, and 54.83% for InstructBlip. These results indicate a clear trend: stronger models show lower unanswerable proportions, implying that more capable models have greater confidence and consequently prefer FI strategy. Additionally, we analyze average unanswerable proportions across four benchmarks—VizWiz, MME, TextVQA, and ScienceQA—and observe a ranking by decreasing uncertainty: VizWiz (58.33%), MME (48.97%), TextVQA (44.20%), and ScienceQA (30.13%). Our findings suggest that benchmarks demanding more detailed perceptual abilities elicit higher uncertainty, whereas simpler visual tasks, such as ScienceQA with cartoon-like images, enable models to respond with higher confidence.

More Analysis of Unanswerable Questions. We further carefully analyze the unanswerable questions along with their associated images, identifying three prevalent scenarios where the model demonstrates increased uncertainty and thus necessitates DT strategy. First, images containing excessive elements, especially those with more than two primary objects, consistently confuse the models, irrespective of whether the style is realistic or cartoon-like (e.g., in ScienceQA). Second, images with unclear or ambiguous regions, notably common in the VizWiz benchmark, directly contribute to heightened uncertainty. Third, dense textual information in images, predominantly observed in TextVQA, visually distracts and overwhelms the VLMs, frequently causing confusion and errors. Additionally, we notice that some questions require specialized external knowledge, particularly within the ScienceQA benchmark, which involves physics or chemistry concepts. Although our FOCUS effectively directs model attention to relevant image regions, the absence of domain-specific knowledge occasionally leads to incorrect predictions. This limitation suggests a promising direction for future work, such as integrating Retrieval-Augmented Generation (RAG) techniques to enrich the model's knowledge base.

Analysis of Confidence Evaluation Strategies. We conducted experiments to further investigate the impact of different confidence evaluation strategies. Specifically, we sampled 100 challenging questions from LLaVA-1.5, consisting of 50 questions incorrectly answered using DT but correctly with FI, and another 50 incorrectly answered using FI without employing DT. We compared three confidence measurement strategies: (1) our original approach (prompt with self-consistency), (2) semantic entropy computed from logits for binary classification of "answerable/unanswerable," and (3) self-consistency combined with semantic entropy. The results are summarized in Table 6. Our analysis reveals several key observations: First, semantic entropy slightly improves performance by addressing the overly stringent requirement of consistent answerable predictions in the original method, reducing unnecessary DT usage on simpler questions. Second, incorporating self-consistency into semantic entropy further enhances performance by averaging entropy scores across multiple predictions, thereby reducing noise and improving robustness.

4 Conclusion

We propose FOCUS, a novel approach that enhances MLLMs in VQA. By adapting thinking strategies to question complexity, FOCUS improves performance and resource efficiency. Experiments show it surpasses existing methods, offering a scalable solution for multimodal reasoning.

5 Limitations

While our FOCUS shows promise in enhancing MLLMs for vision-language tasks, it has several limitations. The method relies on external tools to provide fine-grained visual markers, which could introduce dependencies. Additionally, addressing the fundamental issues of visual hallucinations in MLLMs likely requires modifications at the model architecture and training levels, rather than solely relying on post-processing techniques. Therefore, exploring how to integrate this method more deeply with training processes presents an interesting direction for future research.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 12326612, 62476241), the Natural Science Foundation of Zhejiang Province, China (Grant No. LZ23F020008), and the Zhejiang University-Angelalign Inc. R&D Center for Intelligent Healthcare.

References

Shengnan An, Yifei Li, Zeqi Lin, Qian Liu, Bei Chen, Qiang Fu, Weizhu Chen, Nanning Zheng, and Jian-Guang Lou. 2022. Input-tuning: Adapting unfamiliar inputs to frozen pretrained models. *arXiv preprint arXiv:2203.03131*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee. 2024. Vip-llava: Making large multimodal models understand arbitrary visual prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12914–12923.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yinuo Liu, Yaochen Wang, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. 2024. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. *arXiv preprint arXiv:2402.04788*.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P Xing, and Zhiting Hu. 2022. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *arXiv preprint arXiv:2205.12548*.

Alexey Dosovitskiy. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, et al. 2023. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.

Gemini Team, Google. 2023. Gemini 1 report. Google.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. Ppt: Pre-trained prompt tuning for few-shot learning. *arXiv preprint arXiv:2109.04332*.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617.

Zixi Jia, Jiqiang Liu, Hexiao Li, Qinghua Liu, and Hongbin Gao. 2024. Dcot: Dual chain-of-thought prompting for large multimodal models. In *The 16th Asian Conference on Machine Learning (Conference Track)*.

Songtao Jiang, Yuan Wang, Sibo Song, Yan Zhang, Zijie Meng, Bohan Lei, Jian Wu, Jimeng Sun, and Zuozhu Liu. 2025. Omniv-med: Scaling medical vision-language model for universal visual understanding. *arXiv preprint arXiv:2504.14692*.

Songtao Jiang, Yan Zhang, Ruizhe Chen, Yeying Jin, and Zuozhu Liu. 2024a. Modality-fair preference optimization for trustworthy mllm alignment. *arXiv preprint arXiv:2410.15334*.

Songtao Jiang, Yan Zhang, Chenyi Zhou, Yeying Jin, Yang Feng, Jian Wu, and Zuozhu Liu. 2024b. Joint visual and text prompting for improved object-centric perception with multimodal large language models. *arXiv preprint arXiv:2404.04514*.

Songtao Jiang, Tuo Zheng, Yan Zhang, Yeying Jin, Li Yuan, and Zuozhu Liu. 2024c. Med-moe: Mixture of domain-specific experts for lightweight medical vision-language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3843–3860.

Daniel Kahneman. 2011. *Thinking, fast and slow*. Farrar, Straus and Giroux.

S KAZEMZADE, V Ordonez, M MATTENV, et al. 2014. Referring to objects in photographs of natural scenes. In *Empirical Methods in Natural Language Processing*, volume 28, pages 787–789.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643*.

Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. 2024. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft coco: Common objects in context.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023c. Gpt understands, too. *AI Open*.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. 2024. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14420–14431.

Ron Mokady, Amir Hertz, and Amit H Bermano. 2021. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*.

OpenAI. 2023a. Gpt-3.5 turbo fine-tuning and api updates. https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates.

OpenAI. 2023b. Gpt-4v(ision) system card. https://cdn.openai.com/papers/GPTV_System_Card.pdf. 2023.

R OpenAI. 2023c. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2.

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. Controllable natural language generation with contrastive prefixes. *arXiv preprint arXiv:2202.13257*.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. 2024. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Hao Sun. 2023. Offline prompt evaluation and optimization with inverse reinforcement learning. *arXiv preprint arXiv:2309.06553*.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2021. Spot: Better frozen model adaptation through soft prompt transfer. *arXiv preprint arXiv:2110.07904*.

Junke Wang, Lingchen Meng, Zejia Weng, Bo He, Zuxuan Wu, and Yu-Gang Jiang. 2023. To see is to believe: Prompting gpt-4v for better visual instruction tuning. *arXiv preprint arXiv:2311.07574*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

Qinwei Xu, Xingkun Xu, Chenyi Zhou, Zuozhu Liu, Feiyue Huang, Shaoxin Li, Lifeng Zhu, Zhian Bai, Yuchen Xu, and Weiguo Hu. 2025. Towards normalized clinical information extraction in chinese radiology report with large language models. *Expert Systems with Applications*, 271:126585.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, G. Karypis, and Alexander J. Smola. 2023. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*.

Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibei Yang. 2023. Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. *Advances in Neural Information Processing Systems*, 36:5168–5191.

A Appendix

A.1 Comparison with SoM

Although SoM (Yang et al., 2023) also employs visual prompts to enhance the performance of VLMs, our approach differs significantly in several key aspects:

1. **Scope and Focus**: SoM primarily aims to unleash the visual grounding abilities of VLMs, with its experiments centered on tasks such as Open-vocabulary Image Segmentation, Referring Segmentation, Phrase Grounding, and

Video Object Segmentation. These tasks are inherently focused on visual grounding rather than addressing challenges in the VQA domain. In contrast, we explore the potential of SoM in VQA and propose targeted improvements to adapt it for this domain.

2. **Annotation Strategy**: SoM annotates all content in an image using a SAM-based model (Kirillov et al., 2023). However, our study reveals that this approach is less suitable for VQA tasks, as it fails to highlight critical information and may introduce coarse-grained visual markers that potentially disrupt the semantic integrity of the original image, thereby degrading VQA performance.

3. **Prompting Efficiency**: Current visual or text prompting methods for VQA typically apply prompts uniformly to all questions. Our research demonstrates that not all questions require prompting, and we propose a more cost-efficient solution. This insight provides a new direction for the future development of visual prompting techniques.

B More Analysis and Discussion

B.1 Effect of Chosen Prompt

To evaluate whether our *question complexity evaluation* strategy effectively identifies images that require further fine-grained processing, we used GPT-4V (OpenAI, 2023b). We input a sample of 100 images from VizWiz (Gurari et al., 2018) along with their corresponding questions, using the prompt: "Is this VQA question a complex task requiring fine-grained perception, or a simpler task that only requires coarse-grained perception? Please provide a detailed explanation." After manually reviewing GPT-4V's reasoning, we labeled the images, determining that 59 required fine-grained visual information extraction, while 41 only needed coarse-grained perception. We then explored the accuracy of question complexity evaluation with 1, 3, and 5 response iterations to assess the need for further fine-grained visual information extraction. The accuracies for each iteration were 59%, 81%, and 83%, respectively. These results demonstrate that question complexity evaluation is highly effective in identifying which visual question answering (VQA) questions might confuse the model. This enables us to selectively apply fine-grained information marking to specific images, achieving satisfactory results with just three iterations.

B.2 Effect of Segmentation Model

We also evaluated the segmentation model's ability to mark the extracted keywords. Given that Grounded-SAM (Ren et al., 2024) was trained on a diverse set of open-world objects, and most VQA images come from the COCO dataset (Lin et al., 2015), the marking accuracy is high. We randomly selected 100 samples from VizWiz, achieving an accuracy of over 85%, confirming the effectiveness of our method. However, we set the threshold to 0.7 for the segmentation model. When the model is uncertain about the segmentation, it refrains from making further modifications to the image to avoid disrupting the original visual content. This ensures that the segmentation model does not negatively impact the MLLMs in cases of failure.

B.3 Effect of Keywords Extraction

We randomly selected 100 samples from VizWiz to assess GPT-3.5-Turbo's (OpenAI, 2023a) ability to extract key concepts from VQA questions. Due to GPT-3.5-Turbo's strong information extraction capabilities (Wei et al., 2023) and the relatively simple nature of the VQA questions, 87% of the samples allowed for accurate keyword extraction. However, in the remaining 13%, the relevant objects were not explicitly mentioned in the questions, preventing extraction. In such cases, the segmentation model refrains from adding markers, preserving the image's original semantic content.

Method	ScienceQA(%)
GPT-4V	79.2
Gemini Pro	77.0
GPT-4V + FOCUS	82.4 (+3.2)
Gemini Pro + FOCUS	79.4 (+2.4)

Table 3: Comparative performance of black-box MLLMs on a sample of 250 ScienceQA examples.

Method	ScienceQA (%)	TextVQA (%)	VizWiz (%)	MME
FOCUS	74.4	63.6	58.5	1551.0
Only FI.	71.6	61.3	53.6	1531.3
Only DT.	72.2	61.6	54.9	1538.6

Table 4: Ablation study on Fast Intuition (FI) and DT (Deliberate Thinking). In this experiment, we used LLaVA-1.5 as the MLLM.

Method	SciQA. (%)	TVQA. (%)	VizWiz (%)	MME
SoM	71.3	61.5	54.0	1540.1
FI. + SoM	72.6	61.8	55.2	1544.9
FOCUS (FI. + DT.)	**74.4**	**63.6**	**58.5**	**1551.0**

Table 5: Comparison of Deliberate Thinking (DT) and SoM. SciQA denotes ScienceQA, and TVQA denotes TextVQA.

B.4 Details of Datasets

We evaluate FOCUS using three popular open-source MLLMs and two black-box MLLMs with diverse architectures: LLaVA-1.5 (Liu et al., 2023a), MiniGPT4-V2 (Chen et al., 2023), InstructBLIP (Dai et al., 2023), GPT-4V (OpenAI, 2023b), and Gemini Pro (Gemini Team, Google, 2023). The evaluation is conducted on four datasets specifically chosen to assess distinct capabilities:

ScienceQA (Lu et al., 2022) primarily features images not from the real world, focusing on testing the logical reasoning capabilities of the models. We explore whether finer-grained visual information could enhance the models' reasoning abilities.

TextVQA (Singh et al., 2019) contains images rich in challenging textual information, primarily evaluating the models' text recognition capabilities. We explore our model's ability to extract challenging fine-grained text information in complex visual scenes.

VizWiz (Gurari et al., 2018) consists mostly of real-life images, providing a comprehensive assessment of the MLLMs' proficiency in processing real-world imagery. This dataset includes images that contain complex visual elements, some of which are not very clear, making it difficult for models to interpret the content.

MME (Fu et al., 2023) is a benchmark designed to evaluate MLLMs, addressing the lack of comprehensive assessment tools by measuring both perception and cognition across 14 diverse subtasks. We explore whether our approach could enhance the overall performance of MLLMs on this dataset.

For ScienceQA, TextVQA, and VizWiz, we assess accuracy, while for MME, we use the total score to evaluate the models' overall ability to understand and process different types of data.

C Related Work

Prompt Optimization in VQA Multimodal large language models (MLLMs) have shown promising capabilities in performing visual question answering (VQA) tasks (OpenAI, 2023b; Xu et al., 2025; Liu et al., 2023b; Chen et al., 2023; Jiang et al., 2024c, 2025, 2024a). Prior research aimed at improving MLLMs for Visual Question Answering (VQA) has primarily focused on gradient-based methods (Li and Liang, 2021; Vu et al., 2021; Gu et al., 2021; Liu et al., 2023c; Mokady et al., 2021; Qian et al., 2022; An et al., 2022) and prompt optimization techniques (Deng et al., 2022; Sun, 2023; Jiang et al., 2024b; Zhang et al., 2023). Among these, the Multimodal Chain of Thought (MM-CoT) method (Zhang et al., 2023) is particularly notable for its ability to integrate visual and textual information within LLMs, achieving superior performance in reasoning tasks. However, this approach incurs higher training costs.

Recently, visual prompt strategies have gained prominence in MLLMs. The Set-of-Mark (SoM) method (Yang et al., 2023) is the first to experiment with applying visual marks as prompts on image inputs for MLLMs like GPT-4V, aiming to enhance their grounding capabilities. SoM has demonstrated state-of-the-art results on datasets such as RefCOCOg and DAVIS2017 (KAZEMZADE et al., 2014; OpenAI, 2023c). However, its application to VQA tasks remains underexplored. Moreover, we observe that SoM often segments the entire image when applied to VQA tasks, which may overlook critical information necessary for accurate answers. In contrast, our approach introduces more precise segmentation in MLLMs, focusing on the most relevant parts of the image to significantly improve VQA performance.

C.1 Case Study

There are more case studies on GPT-4V:

Strategy	DT (Count)	FI (Count)	Corrected (DT/FI)
Original Strategy (Prompt + Self-Consistency)	50	50	0/0
Semantic Entropy	41	59	3/9
Self-Consistency Entropy	43	57	4/7

Table 6: Comparison of different confidence evaluation strategies on a subset of 100 challenging cases.

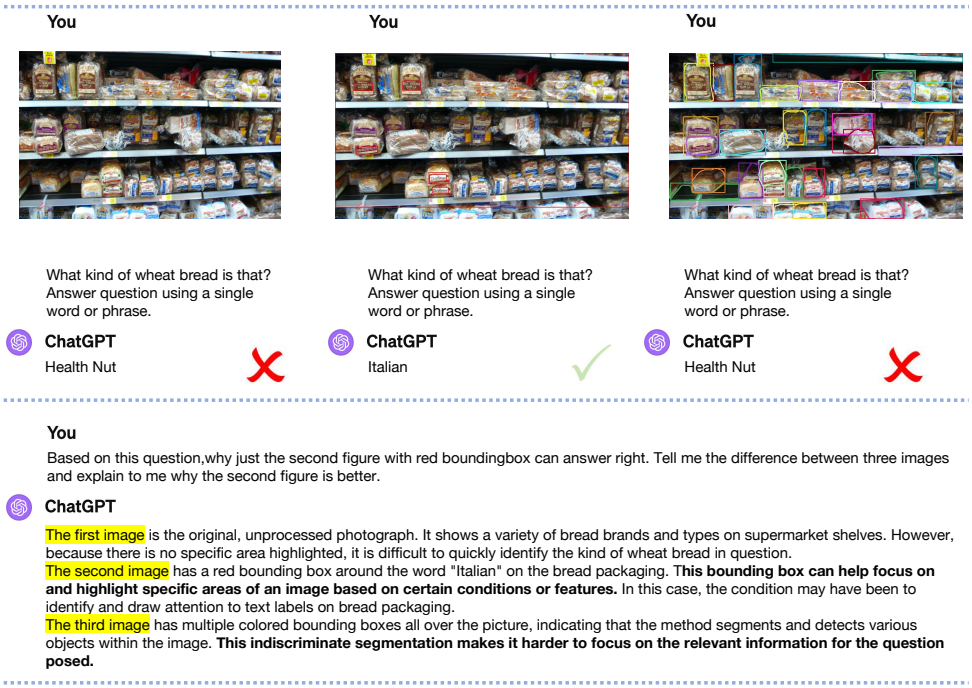


Figure 5: More cases on GPT-4V

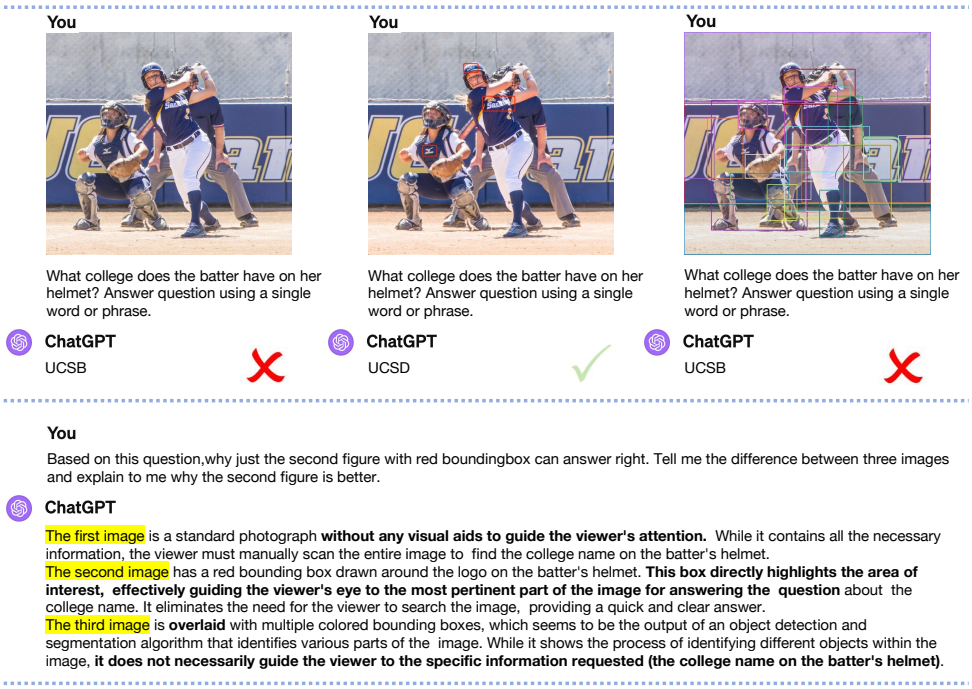


Figure 6: More cases on GPT-4V

Can Community Notes Replace Professional Fact-Checkers?

Nadav Borenstein* **Greta Warren*** **Desmond Elliott** **Isabelle Augenstein**
University of Copenhagen
nb@di.ku.dk grwa@di.ku.dk
de@di.ku.dk augenstein@di.ku.dk

Abstract

Two commonly employed strategies to combat the rise of misinformation on social media are (i) fact-checking by professional organisations and (ii) community moderation by platform users. Policy changes by Twitter/X and, more recently, Meta, signal a shift away from partnerships with fact-checking organisations and towards an increased reliance on crowdsourced community notes. However, the extent and nature of dependencies between fact-checking and *helpful* community notes remain unclear. To address these questions, we use language models to annotate a large corpus of Twitter/X community notes with attributes such as topic, cited sources, and whether they refute claims tied to broader misinformation narratives. Our analysis reveals that community notes cite fact-checking sources up to five times more than previously reported. Fact-checking is especially crucial for notes on posts linked to broader narratives, which are *twice* as likely to reference fact-checking sources compared to other sources. Our results show that successful community moderation relies on professional fact-checking and highlight how citizen and professional fact-checking are deeply intertwined.

1 Introduction

The proliferation of misinformation on social media (Arnold, 2020; Diakopoulos, 2020), along with the rise of generative AI (Augenstein et al., 2024) have led to increasing concerns about people's ability to access trustworthy and credible information, leading to potential harms to public health (Clemente et al., 2022), democracy, and political stability (Reglitz, 2022). Fact-checkers play a crucial role in combatting misinformation (Graves, 2017), and in recent years, have partnered with social media platforms, e.g., Meta, YouTube, and TikTok, to tackle its spread on these platforms. However, due to the scale of misleading content shared online, community moderation (e.g., options to flag potential misinformation, group/server moderators) is often employed in parallel (Morrow et al., 2022), as a complementary approach (e.g., (Google, 2025; TikTok, 2025); see also the practice of *snoping* (Pilarski et al., 2024)). The expansion of fact-checking projects in the last decade (Lauer and Graves, 2024), alongside their broader initiatives to curb misinformation (e.g., citizen media literacy programmes (Juneja and Mitra, 2022)) have been aided by partnerships with social media platforms such as Meta and Google (Graves and Anderson, 2020), which fund independent fact-checking agencies to fact-check potentially false claims on their platform.[1] However, political pressure and accusations of bias and censorship, and most recently, Meta's announcement of its plans to end its partnerships with fact-checkers in the U.S. and implement a community moderation model (Meta, 2025), threatens the financial stability of fact-checking organisations, and hence, their ability to keep up with the increasing volume and sophistication of

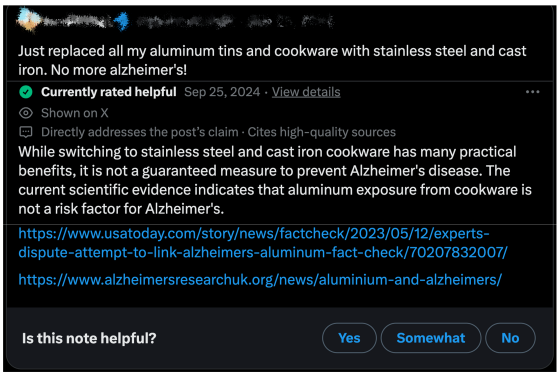


Figure 1: An example of a community note. Notice the fact-checking link and rating.

* Equal contribution.

[1] Fact-checkers provide a judgment of claim veracity and exert no influence on the platforms' content moderation policies (Catalanello and Sanders, 2025).

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 535–552
July 27 - August 1, 2025

misinformation spread (Stencel et al., 2024; IFCN, 2024).

Meta's recent policy shift also implies that these two strategies (fact-checking and community notes) are independent and in opposition, rather than two complementary strategies of tackling online misinformation. In this paper, *we examine Twitter/X community notes as a case study to understand how fact-checking is used in community notes*. Specifically, we investigate the following two questions: **(RQ1) To what extent do community notes rely on the work of professional fact-checkers?** and **(RQ2) What are the traits of posts and notes that rely on fact-checking sources?** Studying the relationship between fact-checking and community notes is vital for understanding the shared role of expert and citizen-driven fact-checking in the global information ecosystem.

We find that at least 1 in 20 community notes rely explicitly on the work of professional fact-checkers, while this reliance is higher still for high-stakes topics such as health and politics. Our experiments also show that fact-checking is vital for debunking misleading content linked to broader narratives or conspiracy theories. These findings imply that high-quality community notes cannot be produced independently of professional fact-checking. They further suggest that the pressure on fact-checkers exerted by platforms and politicians by defunding and discrediting fact-checking organisations will have corrosive effects on the quality of notes and destructive implications for information integrity more widely.

2 Background

Due to space constraints, only the most relevant related work is provided here. Additional relevant studies and background can be found in App. A.

2.1 Community notes

Community moderation has been proposed as a means of addressing the scalability (Martel et al., 2024) and cross-partisanship trust (Flamini, 2019) challenges associated with fact-checking. Twitter/X's Community Notes programme (piloted in 2021 and publicly launched in October 2022 (Twitter/X, 2021)) is a notable example of such a system. Any platform user may volunteer as a Community Notes contributor, although they must achieve a particular 'rating impact score' before they can write notes (Twitter/X, 2024b). Notes that achieve a 'helpful' rating appear underneath the post, explaining why the post is misleading (see Fig. 1). To be rated 'helpful', a note must receive similar levels of helpfulness rating from users with diverse viewpoints (Twitter/X, 2024a).

2.1.1 Characteristics of Community Notes

A small but growing body of work has analysed Twitter/X's Community Notes dataset, focusing on the targets, sources, and limitations of notes.

Targets of notes. Community notes tend to focus on misleading posts from large accounts (Pilarski et al., 2024), focusing on posts that lack important content or present unverified claims as facts (Pröllochs, 2022; Drolsbach and Pröllochs, 2023).

Sources in notes. Analyses have showed that notes were rated more helpful if they link to 'trustworthy' sources and that the majority of sources cited by notes were 'trustworthy' left-leaning news outlets (Pröllochs, 2022). A recent study finds that 55% of URLs used in notes were related to news websites, 18% to research, 9% to social media, 9% to encyclopedic sources, but *just 1%* to fact-checking sources (Kangur et al., 2024).

Limitations of notes. Only 11% of submitted notes reach 'helpful' status (i.e., shown to users) by achieving a cross-perspective (Renault et al., 2024; Wirtschafter and Majumder, 2023), and the time frame for notes to reach the algorithm's required agreement level (15.5 hours on average) limits its capacity to halt misinformation spread (Renault et al., 2024). Posts related to partisan issues are particularly affected by these challenges (Allen et al., 2022). Additional concerns about the notes' efficacy highlight their indifference to the expertise needed for certain claims and reliance on subjective helpfulness rather than objective facts, free labour and inadequate support and guardrails regarding explicit content (Gilbert, 2025).

Our work provides novel insights into the targets, sources and limitations of community notes by shedding light on the relationship between notes and professional fact-checking. Closest to our work is a recent analysis of community notes written in 2024 by the fact-checking organisation Maldita (2025), who also studied the reliance of community notes on professional fact-checkers. They discover that fact-checking organisations are widely used as a reference by notes' authors. The current work provides a more fine-grained analysis by studying the extent to which fact-checking sources form the basis of note-writers' efforts to counter misinfor-

mation and identifying the strategies they employ.

3 Dataset

We download files containing all community notes and their metadata from the official website,[2] which amounts to 1.5M notes authored between January 28th 2021 and January 6th 2025. Of these, a total of 135K are rated by the community as 'Helpful', 51K are rated 'Not helpful', and 1.3M are unpublished, i.e., did not receive enough community ratings to reach a verdict. See Fig. 6 in App. B for statistics.

We filter the notes as follows. First, we remove 526K non-English notes, which we identify by applying the language detection library fastlangdetect.[3] Then, we further filter 268K 'unnecessary' notes—notes attached to tweets that are classified by the community as 'not misleading'. Finally, to focus only on notes that are used to address misinformation, we filter out 44K notes that contain one of the words 'ad', 'spam', or 'phishing'. Following these filtration steps, we are left with a dataset containing 664K notes.

The next step involves categorising the sources that the note authors use to support their claims. First, we use regex to extract all the URLs found in the notes. Importantly, a single note can include multiple external URLs as evidence. See Tab. 2 in App. B for a list of the top-100 most common domains. We classify each URL in our dataset of 664K notes into one of 13 categories (detailed in Fig. 2) using the pipeline described below.

1. Check whether the domain name of the URL is found in a manually curated list of domains of professional fact-checking organisations (See Tab. 3 in App. B for the full list). If so, classify the URL as 'fact-checking'.
2. Otherwise, search for paraphrases of the word 'fact-check' in the URL,[4] and classify it as 'fact-checking' if a match was found.
3. Otherwise, check whether the domain name is found in Tab. 2, which the authors of this paper manually annotated.
4. Otherwise, use GPT-4[5] to classify the domain name into one of the 13 categories. Listing 1 in App. C details the prompt we used.
5. Finally, if GPT-4 fails or outputs an unknown category, label the URL as 'unknown'.

Using this pipeline, we successfully classify 95% of the URLs to one of the 13 categories.

Moreover, we further subsample the notes for performing the in-depth analysis required for answering RQ2 (§4.2). From the notes rated as 'helpful', we sample 3.5K notes with a 'fact-checking' source and a random sample of 22K additional notes. We then used web crawling to scrape the text of the posts to which these notes were attached. We name this subset $\mathcal{S}_{\text{text}}$ for simplicity.

4 Analysis

We analyse the dataset prepared in §3 to answer the two research questions defined in §1.

4.1 RQ1: To what degree do community notes rely on fact-checkers?

According to Fig. 2.a at least 5% of all English community notes contain an external link to professional fact-checkers. This number grows to 7% when only considering notes rated as 'helpful' (Fig. 2.b). Conversely, only 1% of notes rated as 'not helpful' contain a fact-checking source (Fig. 2.c). These figures are significantly larger than what was reported in some previous studies (1.2% (Kangur et al., 2024)), possibly because Kangur et al. (2024) utilise a smaller dataset of fact-checking agencies and classify fact-checking divisions of popular journals as 'news'. The results imply that notes incorporating fact-checking sources are generally considered more helpful.

We further assess whether notes with fact-checking sources are perceived to be of higher quality by analysing individual user ratings of notes both with and without such sources. Specifically, we collect user ratings for a balanced (i.e., including of a fact-checking source or not) sample of 20K notes rated by at least 50 users, and calculated the average ratings for the notes. As can be seen in Fig. 7 in App. B, community notes with fact-checking sources are generally rated higher than their counterparts. Interestingly, while notes with fact-checking links are more likely to be regarded as having a good source (higher *HelpfulGoodSources*), they are also more likely to be rated as *notHelpfulSourcesMissingOrUnreliable*. Tab. 5 in App. B contains a sample of such notes.

[2] https://communitynotes.x.com/guide/en/under-the-hood/download-data

[3] https://github.com/LlmKira/fast-langdetect

[4] These URLs mostly link to the fact-checking divisions of news outlets, e.g., https://apnews.com/article/fact-checking-909101991741.

[5] Version gpt-4o-2024-08-06.

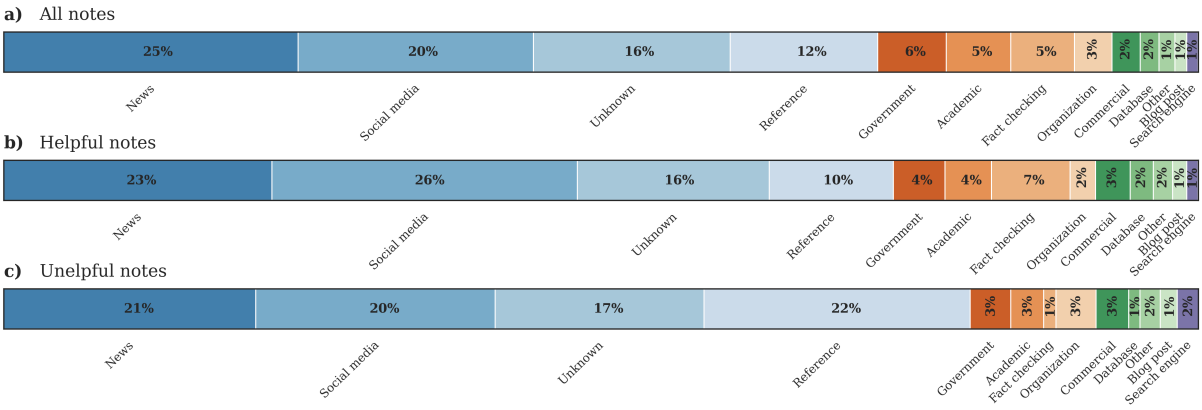


Figure 2: The categories of links used by Community notes' authors as a source. a) all community notes; b) Community notes rated as 'helpful'; c) community notes rated as 'unhelpful'. Notice the 'fact-checking' category.

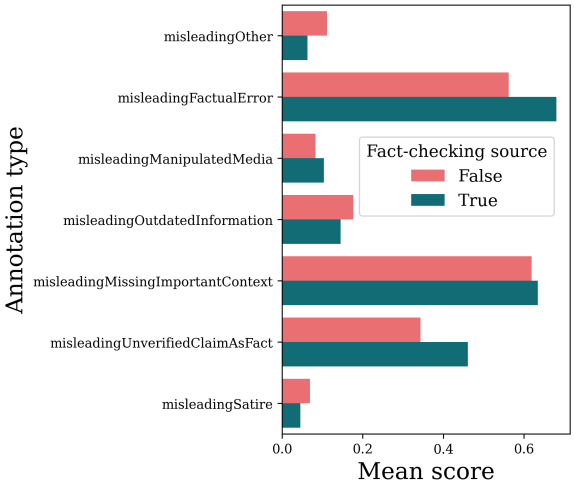


Figure 3: Mean scores of community annotations of misleading posts.

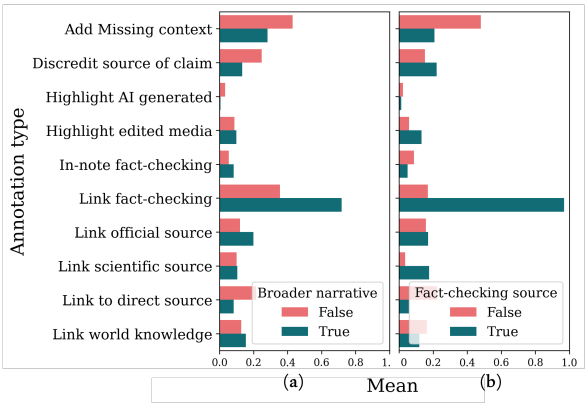


Figure 4: (a) strategies in debunking claims related to broader narratives. (b) the different ways in which fact-checking sources are used to debunk claims.

Conspiracy	FC source ✓	FC source ✗
✓	22%	11%
✗	28%	39%

Table 1: Percentage of samples related to a broader narrative or conspiracy vs. have a fact-checking source.

4.2 RQ2: What are the traits of posts and notes that rely on fact-checking sources?

We begin by performing a topic analysis, comparing topics of posts whose notes reference fact-checking sources to those citing other sources. To this end, we apply a strong zero-shot text classification model[6] to our $\mathcal{S}_{\text{text}}$ subset by classifying spans of the form "`Tweet:<POST TEXT>; Note <NOTE TEXT>`" into one of 13 classes. The authors manually evaluated the quality of the classification results and considered it satisfactory, with the model predicting the correct class in 90% of the cases. Most of the incorrect predictions involved the 'technology' category, with sentences such as "*this is a fake image that was created with AI*" incorrectly labelled as 'technology'. Notably (Fig. 5), fact-checking sources are more likely to be included in posts related to high-stakes issues such as health, science, and scams and less likely to be included in posts on tech or sports.

We then analyse annotations (binary attributes explaining the warrant for the note) by community note authors. Fig. 3 contains the full breakdown of annotations for notes with and without fact-checking sources. Notes containing a link to

[6] https://huggingface.co/r-f/ModernBERT-large-zeroshot-v1 with default settings.

fact-checking sources are overrepresented in posts where unverified information is presented as a fact or when the post contains a factual error. Conversely, they are under-represented in posts with outdated information or satirical content. Tab. 4 in App. B contains a sample of such notes.

These results indicate that community note-writers adapt their strategies based on the stakes and scope of the claim, and the depth of research needed to counter misinformation. We hypothesise that they are more likely to rely on external fact-checking when refuting complex or unverifiable claims (Wuehrl et al., 2024), as well as claims related to conspiracy theories[7] or broader narratives[8] which cannot be fully addressed in the scope of a note. Conversely, claims involving misleading media can often be debunked with examples alone, making fact-checking sources unnecessary. To investigate this hypothesis, the authors of this paper manually annotated 400 <post, note> pairs from $\mathcal{S}_{\text{text}}$ with attributes related to the complexity of the claims and how community notes address them. (see App. C.1 for annotation guidelines). The results (Fig. 4.a) support our hypothesis. Claims related to broader narratives or conspiracy theories[9] are much more likely to include a link to a fact-checking source. In contrast, other types of claims are more likely to be addressed by providing missing context or by invalidating the credibility of the claim's source. Additionally, Fig. 4.b depicts the different ways in which fact-checking sources are used to debunk claims. It demonstrates how such sources are rarely used to provide missing context but rather focus on discrediting sources of claims and providing scientific evidence.

We extend the manual annotation to an LLM-based analysis of 8K balanced <post, note> pairs from $\mathcal{S}_{\text{text}}$. We task OpenAI's GPT-4[10] with determining whether a pair relates to a broader narrative or a conspiracy theory. Listing 2 in App. C details the prompt we used. To evaluate model accuracy, two authors independently labelled 100 balanced pairs, achieving an agreement rate of 0.88 and resolving disagreements through discussion. The model attained an F_1 score of 0.85—strong performance for this challenging task. The results (Tab. 1) support our hypothesis: <post, note> pairs related to a broader narrative or conspiracy theory are *twice* as likely to cite fact-checking sources compared to other sources. In contrast, other pairs are nearly 30% less likely to do so. These findings also highlight the prevalence of such claims and further underscore the importance of fact-checking in combating complex misinformation narratives.

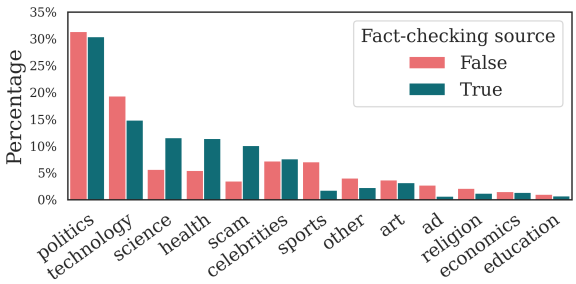


Figure 5: Distribution of notes' topics, with and without a fact-checking source.

5 Conclusion

In this work, we annotate a large corpus of Twitter community notes with attributes such as topic, cited sources, and whether they refute claims tied to broader misinformation narratives. We find that effective community moderation depends on professional fact-checking to an extent far greater than previously reported. We find that community notes linked to broader narratives or conspiracy theories are particularly reliant on fact-checking.

Our results reveal that community notes and professional fact-checking are deeply interconnected—fact-checkers conduct in-depth research beyond the reach of amateur platform users, while community notes publicise their work. The move by platforms to end their partnerships and funding for fact-checking organisations will hinder their ability to fact-check and pursue investigative journalism, which community note writers rely on. This, in turn, will limit the efficacy of community notes, especially for high-stakes claims tied to broader narratives or conspiracies.

Limitations

The main limitations of our work concern the characteristics of the dataset we analyse. First, we restrict our analysis to notes written in English, excluding over half a million notes in other languages.

[7] For example, the claim "Michelle Obama is a male".

[8] These are tweets that perpetuate broader misinformation narratives but do not necessarily contain an explicit conspiracy theory, e.g., the false claim "most immigrants remain firmly dependent on Welfare".

[9] We condense broader narratives and conspiracy theories together for simplicity. Similar trends can be seen when they are analysed separately.

[10] Version `gpt-4o-2024-08-06`.

This decision was made to avoid potential noise and biases arising from the authors' unfamiliarity with public discourse in different regions and reliance on machine translation. In future work, we aim to extend our analysis to other languages.

Moreover, except for a small subset of notes, we did not have access to the original tweets they were written for. Even when the tweet text was available, many contained non-text media, were written in internet vernacular that was challenging to interpret, or lacked important context. These factors limit the accuracy and effectiveness of our models and analysis.

Finally, due to resource constraints, our manual annotation study was limited to a relatively small sample of tweets and notes. In future work, we wish to utilise crowd workers to not only annotate a larger dataset but also increase the diversity and perspective of the annotators.

Broader Impact and Ethical Considerations

Community notes have been proposed as a replacement for professional fact-checkers and a salve to some of the issues encountered by fact-checking. However, our findings support the view that neither community notes nor professional fact-checkers alone are sufficient to combat the spread of misinformation on social media. Rather, a combination of these two strategies could prove a much more effective approach to addressing the full range of false content shared online, for example, by leveraging community notes to identify new checkworthy claims for professional fact-checkers, or relying on fact-checkers' expertise to resolve disputed unpublished notes (see Augenstein et al. (2025) for further recommendations in this vein). In particular, as discussed above, professional fact-checking organisations are especially vital for verifying claims related to broader narratives and conspiracy theories. Moreover, professional fact-checkers remain the only viable strategy currently available for addressing partisan issues, where community notes fall short. Finally, we highlight the potential for incorporating automated, human-in-the-loop fact-checking models to assist professional and community fact-checkers alike in reckoning with vast amounts of both human- and machine-generated content.

Given that this work analyses real-world posts, ethical concerns may arise from using this data for research purposes. Posts from non-protected accounts and Community Notes on Twitter/X are publicly available, however, we acknowledge that they may contain sensitive personal information. To minimise any breach of anonymity and privacy, we anonymised links to individual accounts, and we do not publicly release this information. We do not analyse the posts or notes by individual users, and instead examine aggregated data in the form of topics and sources cited.

Although the Community Notes dataset represents attempts to curb harmful misinformation and conspiracies, given the intense partisanship involved (Allen et al., 2022; Draws et al., 2022), as well as the explicit content of some claims, some instances may be considered offensive. We also acknowledge that our own perspectives and biases as authors shape the impact of our findings in certain ways. For example, as mentioned in the previous section, we were unable to analyse non-English posts in-depth, so our conclusions are likely somewhat focused on discourse in the Anglosphere (e.g., the US, UK, Ireland, Canada, Australia, New Zealand etc.). Furthermore, although we based our criteria for conspiracy theories on well-established sources, e.g., AP News, FactCheck.org, the European Commission, and identified conspiratorial narratives from both left- and right-wing sources, our own perspectives (i.e., as scientists from Western countries) may also have impacted what we considered to be conspiracy theories.

Acknowledgements

This research was co-funded by the European Union (ERC, ExplainYourself, 101077481), by the European Union's Horizon 2020 research and innovation program under grant agreement No. 101135671 (TrustLLM), and by the Pioneer Centre for AI, DNRF grant number P1.

References

Jennifer Allen, Cameron Martel, and David G. Rand. 2022. Birds of a Feather Don't Fact-Check Each Other: Partisanship and the Evaluation of News in Twitter's Birdwatch Crowdsourced Fact-Checking Program. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Phoebe Arnold. 2020. The Challenges of Online Fact Checking: How Technology Can (and Can't) Help. Technical report, Full Fact.

Isabelle Augenstein, Michiel Bakker, Tanmoy Chakraborty, David Corney, Emilio Ferrara, Iryna Gurevych, Scott Hale, Eduard Hovy, Heng Ji, Irene Larraz, Filippo Menczer, Preslav Nakov, Paolo Papotti, Dhruv Sahnan, Greta Warren, and Giovanni Zagni. 2025. Community moderation and the new epistemology of fact checking on social media.

Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, Eduard Hovy, Heng Ji, Filippo Menczer, Ruben Miguez, Preslav Nakov, Dietram Scheufele, Shivam Sharma, and Giovanni Zagni. 2024. Factuality Challenges in the Era of Large Language Models and Opportunities for Fact-Checking. *Nature Machine Intelligence*, pages 1–12.

Rebecca Catalanello and Katie Sanders. 2025. Meta is Ending Its Third-Party Fact-Checking Partnership with US Partners. Here's How That Program Works.

Yuwei Chuai, Moritz Pilarski, Gabriele Lenzini, and Nicolas Pröllochs. 2024a. Community Notes Reduce the Spread of Misleading Posts on X.

Yuwei Chuai, Anastasia Sergeeva, Gabriele Lenzini, and Nicolas Pröllochs. 2024b. Community Fact-Checks Trigger Moral Outrage in Replies to Misleading Posts on Social Media. ArXiv:2409.08829 [cs].

Yuwei Chuai, Haoye Tian, Nicolas Pröllochs, and Gabriele Lenzini. 2024c. Did the Roll-Out of Community Notes Reduce Engagement With Misinformation on X/Twitter? *Proceedings of the ACM on Human-Computer Interaction*, 8(CSCW2):1–52.

Suárez Vicente Javier Clemente, Eduardo Navarro-Jiménez, Juan Antonio Simón-Sanjurjo, Ana Isabel Beltran-Velasco, Carmen Cecilia Laborde-Cárdenas, Juan Camilo Benitez-Agudelo, Álvaro Bustamante-Sánchez, and José Francisco Tornero-Aguilera. 2022. Mis–Dis Information in COVID-19 Health Crisis: A Narrative Review. *International Journal of Environmental Research and Public Health*, 19(9).

Nicholas Diakopoulos. 2020. Computational News Discovery: Towards Design Considerations for Editorial Orientation Algorithms in Journalism. *Digital journalism*, 8(7):945–967.

Tim Draws, David La Barbera, Michael Soprano, Kevin Roitero, Davide Ceolin, Alessandro Checco, and Stefano Mizzaro. 2022. The Effects of Crowd Worker Biases in Fact-Checking Tasks. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2114–2124, Seoul Republic of Korea. ACM.

Chiara Patricia Drolsbach and Nicolas Pröllochs. 2023. Diffusion of Community Fact-Checked Misinformation on Twitter. *Proceedings of the ACM on Human-Computer Interaction*, 7(CSCW2):1–22.

Chiara Patricia Drolsbach, Kirill Solovev, and Nicolas Pröllochs. 2024. Community Notes Increase Trust in Fact-Checking on Social Media. *PNAS Nexus*.

Daniela Flamini. 2019. Most Republicans Don't Trust Fact-Checkers, and Most Americans Don't Trust the Media.

Yang Gao, Maggie Zhang, and Huaxia Rui. 2024. Can Crowdchecking Curb Misinformation? Evidence from Community Notes.

Sarah Gilbert. 2025. Three Reasons Meta Will Struggle with Community Fact-Checking. *MIT Technology Review*.

Google. 2025. Misinformation Policies - Youtube Help.

Lucas Graves. 2017. Anatomy of a Fact Check: Objective Practice and the Contested Epistemology of Fact Checking. *Communication, culture & critique*, 10(3):518–537.

Lucas Graves and C.W. Anderson. 2020. Discipline and Promote: Building Infrastructure and Managing Algorithms in a "Structured Journalism" Project by Professional Fact-Checking Groups. *New Media & Society*, 22(2):342–360.

Jordyn Haime. 2022. Taiwan's Amateur Fact-Checkers Wage War on Fake News from China.

IFCN. 2024. State of the Fact-Checkers Report. Technical report, International Fact-Checking Network.

Prerna Juneja and Tanushree Mitra. 2022. Human and Technological Infrastructures of Fact-Checking. *Proc. ACM Hum.-Comput. Interact.*, 6(CSCW2).

Uku Kangur, Roshni Chakraborty, and Rajesh Sharma. 2024. Who Checks the Checkers? Exploring Source Credibility in Twitter's Community Notes. ArXiv:2406.12444 [cs].

Sarawut Kankham and Jian-Ren Hou. 2024. Community Notes vs. Related Articles: Assessing Real-World Integrated Counter-Rumor Features in Response to Different Rumor Types on Social Media. *International Journal of Human–Computer Interaction*, pages 1–15.

Laurens Lauer and Lucas Graves. 2024. How to Grow a Transnational Field: A Network Analysis of the Global Fact-Checking Movement. *New Media & Society*, 0(0):19.

Maldita. 2025. The Impact of Fact-Checkers in X's Community Notes.

Cameron Martel, Jennifer Allen, Gordon Pennycook, and David G. Rand. 2024. Crowds Can Effectively Identify Misinformation at Scale. *Perspectives on Psychological Science*, 19(2):477–488. PMID: 37594056.

Meta. 2025. More Speech and Fewer Mistakes.

Nicholas Micallef, Vivienne Armacost, Nasir Memon, and Sameer Patil. 2022. True or False: Studying the Work Practices of Professional Fact-Checkers. *Proc. ACM Hum.-Comput. Interact.*, 6(CSCW1).

Garrett Morrow, Briony Swire-Thompson, Jessica Montgomery Polny, Matthew Kopec, and John P Wihbey. 2022. The Emerging Science of Content Labeling: Contextualizing Social Media Content Moderation. *Journal of the Association for Information Science and Technology*, 73(10):1365–1386.

Moritz Pilarski, Kirill Olegovich Solovev, and Nicolas Pröllochs. 2024. Community Notes vs. Snoping: How the Crowd Selects Fact-Checking Targets on Social Media. *Proceedings of the International AAAI Conference on Web and Social Media*, 18:1262–1275.

Nicolas Pröllochs. 2022. Community-Based Fact-Checking on Twitter's Birdwatch Platform. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 16, pages 794–805.

Merten Reglitz. 2022. Fake News and Democracy. *J. Ethics & Soc. Phil.*, 22:162.

Thomas Renault, David Restrepo-Amariles, and Aurore Troussel. 2024. Collaboratively Adding Context to Social Media Posts Reduces the Sharing of False News. HEC Research Papers Series 1519, HEC Paris.

Mohammed Saeed, Nicolas Traub, Maelle Nicolas, Gianluca Demartini, and Paolo Papotti. 2022. Crowdsourced Fact-Checking at Twitter: How Does the Crowd Compare With Experts? In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 1736–1746, New York, NY, USA. Association for Computing Machinery.

Mark Stencel, Erica Ryan, and Joel Luther. 2024. With Half the Planet Going to the Polls in 2024, Fact-Checking Sputters. Technical report, Duke Reporters' Lab.

TikTok. 2025. Testing a New Feature to Enhance Content on TikTok.

Twitter/X. 2021. Introducing Birdwatch, a Community-Based Approach to Misinformation.

Twitter/X. 2024a. Note Ranking Algorithm.

Twitter/X. 2024b. Rating and Writing Impact.

Greta Warren, Irina Shklovski, and Isabelle Augenstein. 2025. Show Me the Work: Fact-Checkers' Requirements for Explainable Automated Fact-Checking. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA. Association for Computing Machinery.

Valerie Wirtschafter and Sharanya Majumder. 2023. Future Challenges for Online, Crowdsourced Content Moderation: Evidence from Twitter's Community Notes. *Journal of Online Trust and Safety*, 2(1).

Amelie Wuehrl, Yarik Menchaca Resendiz, Lara Grimminger, and Roman Klinger. 2024. What Makes Medical Claims (Un)Verifiable? Analyzing Entity and Relation Properties for Fact Verification. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2046–2058, St. Julian's, Malta. Association for Computational Linguistics.

Taha Yasseri and Filippo Menczer. 2023. Can Crowdsourcing Rescue the Social Marketplace of Ideas? *Communications of the ACM*, 66(9):42–45.

Andy Zhao and Mor Naaman. 2023. Insights from a Comparative Study on the Variety, Velocity, Veracity, and Viability of Crowdsourced and Professional Fact-Checking Services. *Journal of Online Trust and Safety*, 2(1).

A Extended Background

A.1 Impact of Community Notes on misinformation spread

Posts identified by community notes as misleading have been found to attain less virality (reposts, quote tweets and replies) than non-misleading posts (Drolsbach and Pröllochs, 2023; Renault et al., 2024). Community notes have also been shown to increase the probability of tweet retractions and deletions and speed up the retraction process (Gao et al., 2024; Renault et al., 2024). However, other studies have found less positive evidence; for example, that users' followers, likes and engagement increase after their post receives a community note (Wirtschafter and Majumder, 2023). Curiously, one study claims that showing community notes on posts reduced the spread of misleading posts by an average of 61% (Chuai et al., 2024a), while a more recent analysis by the same authors found no effect of community notes on engagement with misinformation (Chuai et al., 2024c).

People shown community notes alongside misleading social media posts were more accurate in identifying misleading posts, and the notes were judged to be more trustworthy than context-free misinformation flags (e.g., "Checked by fact-checkers" or "Checked by other social media users"), regardless of (US-centric) political beliefs (Drolsbach et al., 2024). People shown either community notes or related news article suggestions were both less likely to believe and report misleading information compared to a control group: community notes were more effective in reducing belief and sharing intention for positive rumours, while articles were more effective for negative rumours (Kankham and Hou, 2024). On the other hand, displaying community notes leads users to post more negative and angry replies to misleading posts (Chuai et al., 2024b), while crowd workers are also prone to cognitive biases, such as overestimating a statement's truthfulness the more they liked its claimant, and general overconfidence in their ability to ascertain truthful statements (Draws et al., 2022).

A.2 Professional fact-checking and community note practices

Although fact-checks and community notes share similarities in how they address misleading claims, they also differ in key elements of practice and communication (Kankham and Hou, 2024). Fact-checking typically involves the analysis and verification of public claims, and in addition to verifying claims, in recent years many fact-checking organisations have also assumed a wider role in combating misinformation spread, e.g., long-term investigative journalism projects, media literacy programs (Juneja and Mitra, 2022). Professional fact-checkers signatory to the International Fact-Checking Network follow a rigorous set of principles and transparency commitments[11] and a structured workflow: (i) claim selection; (ii) collecting evidence; (iii) deciding on a verdict; and (iv) writing the fact-checking article (Graves, 2017; Micallef et al., 2022; Warren et al., 2025). In contrast, any platform user can contribute to community notes under anonymity, and the rating approach relies on the 'wisdom of crowds', with little oversight or transparency regarding biases of note-writers. Community note writers and fact-checkers tend to target similar topics (e.g., health and politics) (Saeed et al., 2022). Fact-checkers must rely on credible sources and evidence to convince the reader, while note writers often disagree with fact-checkers on what constitutes a reliable source, particularly on political ideological grounds (Saeed et al., 2022). The verdicts reached in notes tend to agree with those of fact-checkers — however, due to political polarisation (Yasseri and Menczer, 2023), community notes on contentious political issues rarely reach a consensus (Saeed et al., 2022). Fact-checking articles, which are subject to multiple rounds of editorial scrutiny, are more formal and standardised in style than community notes, which vary considerably and can employ a range of persuasion techniques, such as appeals to emotion or other logical fallacies (Kankham and Hou, 2024). Moreover, community notes typically serve as direct rebuttals to misleading posts, while fact-checking articles may address a more general claim than is expressed in a specific post. Finally, fact-checking articles are a one-way exchange, while community notes represent a more horizontal and interactive dialogue between writer and recipient of the fact-check (Kankham and Hou, 2024). Prior work has found that collaborative fact-checking, a distinct approach to community notes for its Wikipedia-style approach that allows users to edit, as well as up and downvote user-written posts (Haime, 2022), can produce fact-

[11] https://www.ifcncodeofprinciples.poynter.org/the-commitments

checks with comparable speed, reliability, objectivity, clarity and persuasiveness to those written by professional fact-checks (Zhao and Naaman, 2023). However, laypeople's work is expedited by existing fact-checking articles, and amateurs tend to defer to professional fact-checkers for topics requiring specific expertise, such as medical claims (Zhao and Naaman, 2023). Our work builds on current understanding of the relationship between professional fact-checking and community moderation by examining the extent to which community note writers deploy the work of fact-checkers in their notes.

B Additional Material

This section details additional results or material referenced from the paper's main body.

Fig. 6 A histogram of the number of community notes written every month and their rating (*helpful*, *not helpful*, or *needs more data*).

Fig. 7 Community ratings of notes with and without fact-checking source.

Tab. 2 List of top 100 most common domains found in the community notes dataset, and their categorisation.

Tab. 3 List of professional fact-checking organisations and their URLs.

Tab. 4 A sample of tweets, notes, and their community annotations, as well as whether the note contains a fact-checking link.

Tab. 5 Examples of community notes containing fact-checking sources that are rated as having *notHelpfulSourcesMissingOrUnreliable*.

C Reproducibility

Listing 1 The prompt used to classify URLs into categories.

Listing 2 The prompt used to classify tweets and notes into broader narratives and conspiracy theories.

C.1 Manual Annotation Setup

We annotate 400 (tweet, note) pairs from $\mathcal{S}_{\text{text}}$ with 12 binary attributes. Each (tweet, note) pair was annotated in a multi-label fashion, i.e., more than one attribute can be selected at the same time. Fig. 8 depict our simple annotation setup, with the 12 attributes being as follows.

Broader narrative Whether the (tweet, note) pair is related to a broader narrative or a conspiracy theory.

Discredit source of claim If the community note describes the source shared by the original post as non-credible.

Add missing context If the community note provides some missing context to refute a claim.

Highlight AI generated If the community note claims that the post shared AI-generated content.

Highlight edited media If the community note claims that the post shared some media that was edited (edited with Photoshop, the clip was cut, etc.).

Link to direct source If the community note shares a link to a source where an entity says that a claim made about them is false.

Link official source If the community note shares a link to an official source such as a government website.

Link scientific source If the community note shares a link to some scientific article or website.

Link world knowledge If the community note shares a link to some reference resources such as Wikipedia.

Link fact-checking If the community note shares a link to a professional fact-checking organisation.

In-note fact-checking If the community note performs an in-note fact-check by cross-referencing several sources and constructing a compelling argument.

Domain	Category	Domain	Category
x.com	social media	thehill.com	news
twitter.com	social media	amp.theguardian.com	news
youtube.com	social media	whitehouse.gov	government
youtu.be	social media	news.sky.com	news
un.org	organisation	merriam-webster.com	reference
u.today	news	techarp.com	news
t.co	social media	cbc.ca	news
snopes.com	fact checking	politifact.com	fact checking
en.m.wikipedia.org	reference	pbs.org	commercial
en.wikipedia.org	reference	telegraph.co.uk	news
google.com	search engine	businessinsider.com	news
instagram.com	social media	time.com	news
britannica.com	reference	justice.gov	government
reuters.com	news	cnbc.com	news
bbc.co.uk	news	wsj.com	news
apnews.com	news	sciencedirect.com	academic
bbc.com	news	msn.com	news
nytimes.com	news	statista.com	reference
theguardian.com	news	business.x.com	commercial
vice.com	news	amp.cnn.com	news
usatoday.com	news	congress.gov	government
factcheck.org	fact checking	factcheck.afp.com	fact checking
cnn.com	news	yahoo.com	search engine
washingtonpost.com	news	timesofindia.indiatimes.com	news
ncbi.nlm.nih.gov	academic	thelancet.com	academic
nbcnews.com	news	hrw.org	organisation
help.twitter.com	reference	healthfeedback.org	fact checking
cdc.gov	government	fda.gov	government
npr.org	news	m.youtube.com	social media
forbes.com	news	law.cornell.edu	academic
newsweek.com	news	medium.com	blog post
fullfact.org	fact checking	healthfeedback.org	fact checking
dailymail.co.uk	news	who.int	organisation
cbsnews.com	news	haaretz.com	news
web3antivirus.io	database	axios.com	news
timesofisrael.com	news	mayoclinic.org	commercial
help.x.com	reference	nejm.org	academic
nypost.com	news	scienceexchange.caltech.edu	academic
aljazeera.com	news	indiatoday.in	news
reddit.com	social media	bloomberg.com	news
independent.co.uk	news	pewresearch.org	academic
usgs.gov	academic	jamanetwork.com	academic
abcnews.go.com	news	leadstories.com	news
nature.com	academic	dictionary.cambridge.org	reference
gov.uk	government	jpost.com	news
web.archive.org	database	archive.ph	database
foxnews.com	news	healthline.com	commercial
tiktok.com	social media	abc.net.au	news
edition.cnn.com	news	france24.com	news

Table 2: List of top 100 most common domains found in the community notes dataset, and their categorisation.

Name	URL	Language	Region/domain
Lead stories	leadstories.com	English	Global
AFP Factuel	factuel.afp.com	French	Global
AAP FactCheck	aap.com.au/factcheck	English	Australia
Full Fact	fullfact.org	English	Global
Science Feedback	science.feedback.org	English	Science
Politifact	politifact.com	English, Spanish	USA
HoaxEye	hoaxeye.wordpress.com	English	Images
Logically Facts	logicallyfacts.com	Multiple	Europe/India
FactCheckNI	factcheckni.org	English	North Ireland
DFRLab	dfrlab.org	English	Global
FactReview	factreview.gr	Greek	Global
Lupa	lupa.uol.com.br/jornalismo	Portuguese	Global
Check your fact	checkyourfact.com	English	Global
Climate feedback	climatefeedback.org	English	Climate
Factcheck	factcheck.org	English	USA
Health feedback	healthfeedback.org	English	Health
Snopes	snopes.com	English	US
aosfatos	aosfatos.org	Portuguese	Global
Demagog	demagog.org.pl/fake_news	Polish	Poland
FakeReporter	fakereporter.net	Hebrew	Israel
litmus factcheck	litmus-factcheck.jp	Japanese	Japan
Climate Feedback	climatefeedback.org	English	Global
AFP	factcheck.afp.com	English	Global
USA Today	usatoday.com/story/news/factcheck	English	USA
Statesman	statesman.com	English	USA
Dallas News	dallasnews.com/news/politifact	English	USA
Google Fact Check	toolbox.google.com/factcheck	English	Global
MediaBias/FactCheck	mediabiasfactcheck.com	English	Global
MedDMO	meddmo.eu	English, Greek	Greece, Cyprus, Malta
Poynter	poynter.org/fact-checking	English	USA
Newsmeter	newsmeter.in/fact-check	English, Tamil	India
Africa Check	africacheck.org	English	Africa
Fact Crescendo India	english.factcrescendo.com	English	India
Factseeker	factseeker.lk	English	Sri Lanka
Fact Crescendo Thailand	thailand.factcrescendo.com	Thai	Thailand
Fact Crescendo Afghanistan	afghanistan.factcrescendo.com	Persian	Afghanistan
Only Fact	onlyfact.in	English	India
Factly	factly.in	English	India
Fact Crescendo Sri Lanka	srilanka.factcrescendo.com	Sinhala	Sri Lanka
Fact Crescendo Cambodia	cambodia.factcrescendo.com	Cambodian	Cambodia
Becid	becid.eu	Baltic langs	Baltic
Fact Hunt	facthunt.in	English	India
Tec Arp	techarp.com	English	Global (based in Malaysia)
10 news	10news.com/news/fact-or-fiction	English	USA
RMIT Fact Check	rmit.edu.au	English	Australia
Gigafact	gigafact.org	English	USA
Ayupp	ayupp.com/fact-check	English	India
The Journal	thejournal.ie	English	Ireland

Table 3: List of professional fact-checking organisations and their URLs.

Tweet	Note	misleadingUnverifiedClaimAsFact	misleadingOutdatedInformation	misleadingFactualError	misleadingSatire	Fact Checking source
The NASA War Document is absolutely terrifying https://t.co/...	misrepresenting a presentation by NASA scientist Dennis Bushnell, The lecture was not detailing plans by NASA to attack the world it was a lecture for defense industry professionals, and how defense tactics might rise to meet evolving threats in the future. https://leadstories.com/hoax-alert/2021/06/fact-check-the-future-is-now-is-not-a-nasa-war-document-plan-for-world-domination-and-phasing-out-of-humans.html	✓	✗	✗	✗	✓
BREAKING NEWS: International Criminal Investigation calls on every public citizen to recommend indictments for Bill Gates, Anthony Fauci, Pfizer, BlackRock, Tedros and Christian Drosten for pushing everyone to receive the ineffective highly dangerous lethal experimental vaccines...	Video has been fact-checked by USA Today, was found to be misleading, and promotes a conspiracy theory about COVID ... https://ca.movies.yahoo.com/movies/fact-check-viral-video-promotes-204414488.html	✓	✗	✗	✗	✓
1) California is RED. It is just because of the MASSIVE Election Fraud that stupid, brainwashed people believe Calif. is blue. Joe Biden won only in the SFO Bay area ...	The map shows the results of Reagan's reelection in 1984, not Biden's election in 2020. https://en.wikipedia.org/wiki/1984_United_States_presidential_election_in_California	✗	✓	✗	✗	✗
Davis blows up $100,000 fireworks in Kai Cenat setup During the Mr Beast Stream ...	The second photo is from a house fire in Atlanta in 2019. https://www.11alive.com/article/news/local/woodland-brook-drive-cause-of-house-fire/85-ecb7df9b-5f65-44e9-bf9d-8c162d36c334	✗	✓	✗	✗	✗
@cnviolations I swear community notes are the only good thing Elon added since he bought Twitter.	Community notes was first launched under former Twitter CEO Jack Dorsey in 2021 under the name of 'Birdwatch'. The only thing Elon Musk did was that he renamed the feature to community notes. https://blog.twitter.com/en_us/topics/product/2021/introducing-birdwatch-a-community-based-approach-to-misinformation https://www.reuters.com/article/factcheck-elon-birdwatch-idUSL1N31Z2VG/	✗	✗	✓	✗	✓
Thailand will become the first country to make the contract null and void, meaning that Pfizer will become responsible for all vaccine injuries ...	Thailand has no plans to void its Pfizer COVID vaccine contract, an official with the country's National Vaccine Institute said. Thailand's Department of Disease Control also rejected the claims as 'fake news.' ... https://apnews.com/article/fact-check-covid-vaccine-pfizer-thailand-203948163859	✗	✗	✓	✗	✓
Hilarious tweets by footballers, A thread: 1. Virgil Van Dijk [Current Liverpool Captain] https://t.co/...	Virgil Van Dijk did not tweet this, the tweet was made by a fan account in his name. https://www.pinkvilla.com/sports/fact-check-did-virgil-van-dijk-really-root-for-man-u-because-no-one-likes-liverpool-in-resurfaced-viral-tweet-1287250	✗	✗	✗	✓	✓
Rob Reiner announces he's on the Epstein Client List and Epstein Flight logs. What a fool! When a lawyer tells me to STFU, I STFU! https://t.co/...	This is a digitally altered photo that might be misinterpreted even if used as a joke. The name Rob Reiner is misspelled, and the text is not on Reiner's X timeline. https://twitter.com/robreiner?t=iqu43-NszIW5oOM_KqRSpw	✗	✗	✗	✓	✗

Table 4: A sample of tweets, notes, and their community annotations, as well as whether the note contains a fact-checking link.

ID	summary
0	This claim ruled mostly false. https://www.politifact.com/factchecks/2020/may/07/facebook-posts/facebook-post-cites-doctors-widely-disputed-calcul/
1	The RedState article claims "the shots do not stop transmission of the virus. This is false. ""Vaccines provide significant protection from 'getting it' – infection – and 'spreading it' – transmission – even against the delta variant." Source: https://www.usatoday.com/story/news/factcheck/2021/11/17/fact-check-covid-19-vaccines-protect-against-infection-transmission/6403678001/
2	There is no proof of this, the photo is real, it's not the last photo of the child. But snoops say there is a tenuous link the parents used the same law firm to represent them as Maxwell https://www.snopes.com/fact-check/ghislaine-maxwell-jonbenet-ramsey/
3	unfounded https://www.snopes.com/fact-check/ashley-biden-diary-afraid/
4	The mRNA vaccine does not cause cancer: https://www.factcheck.org/2024/05/still-no-evidence-covid-19-vaccination-increases-cancer-risk-despite-posts/
5	Many of the details in this popular essay are inaccurate and too numerous to list here. The essay was fact checked by Snopes in 2005: https://www.snopes.com/fact-check/the-price-they-paid/
6	POLITIFACT - rates False. The report analysed a small sample of 128 temp stations out of several thousand volunteer-run stations, then extrapolated results. NOAA uses 2 programs to record daily temps. The report did not look at the 900 more sophisticated automated stations. https://www.politifact.com/factchecks/2022/aug/19/facebook-posts/fact-checking-talking-point-about-corrupted-climat/
7	There is no verifiable evidence of campaign espionage in either the 2020 or the 2016 presidential elections. https://www.snopes.com/fact-check/obama-spying-trump-campaign/ https://www.washingtonpost.com/politics/2019/05/06/whats-evidence-spying-trumps-campaign-heres-your-guide/
8	Ladapo did get caught altering COVID vaccine study findings. Ladapo replaced the language from an earlier study draft that found no significant risk from COVID vaccines, to then state there was a high risk https://healthfeedback.org/claimreview/analysis-florida-department-health-surgeon-general-joseph-ladopo-contains-multiple-methodological-problems-covid-19-mrna-vaccines/ https://healthexec.com/topics/clinical/COVID-19/florida-surgeon-general-altered-covid-19-study-findings

Table 5: Examples of community notes containing fact-checking sources that are rated as having *notHelpfulSourcesMissingOrUnreliable*.

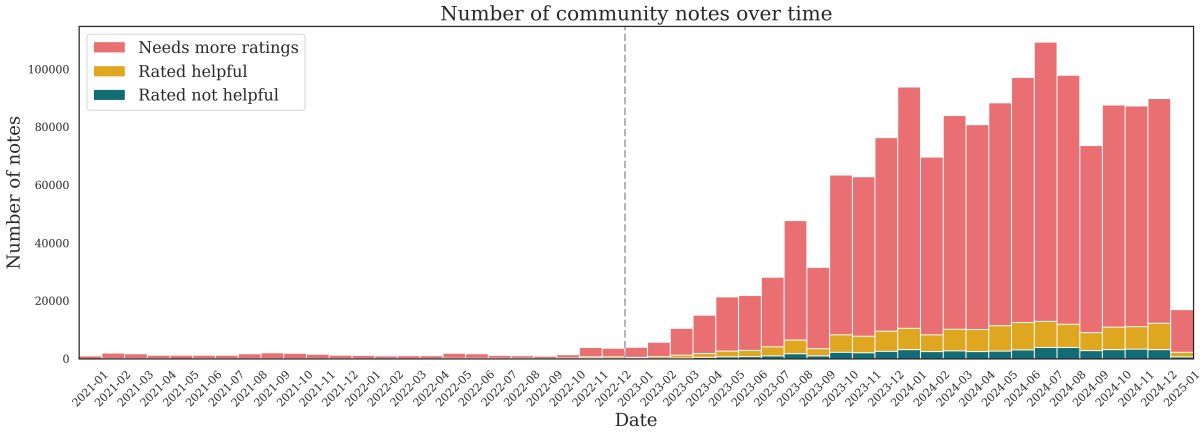


Figure 6: A histogram of the number of community notes written every month and their rating (*helpful*, *not helpful*, or *needs more data*. The grey vertical line (December 2022) indicates the date when the community notes became visible worldwide.

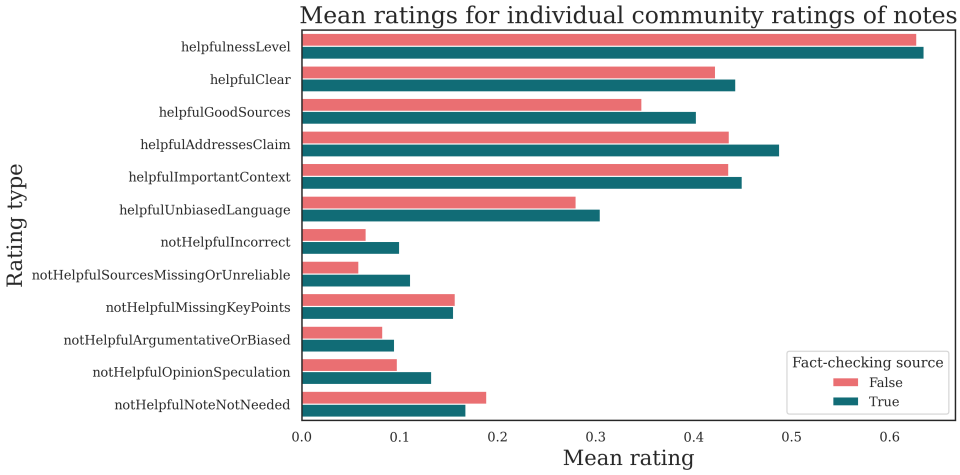


Figure 7: Community ratings of notes with and without fact-checking source.

```
SYSTEM PROMPT
You are a professional IT system who has a vast knowledge of the internet and its
    content. Your goal is simple, but very important: Classify URLs into
    categories. Choose only from the provided categories!


USER PROMPT
Read the following URLs.
Your goal is to categorize each url into one of the pre-defined categories.

Chose from the following list of categories:
Categories =
[
    "social media",  # Social media sites like Facebook, Twitter, Youtube etc.
    "news",  # Websites of news outlets or other organisations that report current
    events, such as the nytimes, the guardian, etc.
    "government",  # Government agencies and organisations, as well as websites
    related to policies and guidelines,  such as the CDC, department of education,
    FDA, etc.
    "academic",  # Academic sources, journals, and magazines, such as pubmed,
    nature, sciencedirect, etc.
    "blog post",  # Independent blog posts about various topics, including cooking,
    travel, home improvement, fandom, reviews, etc.
    "fact checking",  # professional fact checking organisations
    "database",  # Public databases such as google drive, archive.com, dropbox, etc.
    "commercial",  # Webpages of commercial organisations such as BMW, Delta, Nike,
    etc.
    "reference",  # Public resources such as encyclopedias, dictionaries, advocacy
    sources, guides, DIYs, statistics, religious sources, travel information, usage
    guidelines, Q&As, terms of services, etc.
    "organisation",  # non-commercial and non-government organisations such as WHO,
    the UN, Greenpeace, LA-Lakers, etc.
    "other",  # Any other website that does not fit into one of the previous
    categories.
    "unknown",  # if it is impossible to determine the category of the webpage.
]

Output format example:
[
    {
        id: <ID>,
        url: <URL>,
        category: <CATEGORY>,
    }
]


URLs:
<URLS>
```

Listing 1: The prompt used to classify URLs into categories.

```
SYSTEM PROMPT
You are a professional fact-checker who specializes in analyzing misinformation
    spread on social media.
Your goal is to analyse a tweet and a community note written about the tweet and
    decide whether the tweet spread misinformation related to a known conspiracy
    theory or a misleading wider narrative, and if so, which one is it.



USER PROMPT
Read the following tweets and community notes written about them.\nYour goal is to
    analyse them and decide whether each tweet spread misinformation related to a
    known conspiracy theory or a similar misleading wider narrative, and if so (and
    only if so!), which one.
Include your reasoning. Output the results as a json file. If a tweet does not
    relate to a conspiracy theory or a misleading wider narrative, output "none" in
    the json.

- Tweets *do not* discuss a wider narrative if the misleading information is tied
    to a specific singular event that is not connected to major topics on the
    public discourse.
They do discuss a wider narrative if the misleading information is tied to a known
    conspiracy theory or to major topics on the public discorse.

Chose from the following list of theories and wider narrative:
CONSPIRACY_THEORIES =
[
    September 11,
    October 7,
    the great replacement,
    COVID was intentionally spread,
    the COVID outbreak is fake,
    2020 election fraud,
    vaccines cause autism,
    5G towers,
    Russian invasion of Ukraine,
    flat earth,
    chemtrails,
    Q-Anon and deep state,
    Epstein files,
    Barack Obama was not born in the USA,
    Michelle Obama is a man,
    LGBT grooming,
    fluorite in the water,
    climate change,
    Holocaust denial,
    Hunter Biden and Ukraine,
    other,
]

Output format example:
[
    {
        id: <ID>,
        is_related_to_conspiracy: <True/False>,
        conspiracy: <CONSOIRACY (or None)>,
        reasoning: <REASONING>\
    }
]

    Tweets and notes:
    <TWEETS_AND_NOTED>
```

Listing 2: The prompt used to classify tweets and notes into broader narratives and conspiracy theories.

	E	F	K	L	M	N	O	P	Q	R	S	T	U	V
1	**Original tweet**	**Note**	**Broader narrative**	**Conspiracy theory**	**Unreliability of claim**	**Link to missing context (e.g. the original context)**	**Highlight AI generated content**	**Highlight manipulated media (altered, edited, staged, etc.), link to original media**	**Link to direct source**	**Link to official source (e.g. government info)**	**Link to scientific source**	**World knowledge e.g. Wikipedia**	**Link to investigate journalism (fact-checking, news)**	**In-note fact-checking: multiple evidence + argumentation**
2	IS THAT A BAT? A MOUSE? WHAT?🔒🔒 https://t.co/EGau0iuwbi	Turkey neck. They put it inside the turkey. https://www.ta	☑	☐	☐	☑	☐	☐	☐	☐	☐	☐	☐	☐
3	BREAKING: Armed communist Maduro gangs are currently storming p They aim to steal the ballot boxes. https://t.co/L1vbQQF5Z7	The video is showing the theft of air-conditioning units, not	☑	☑	☐	☐	☐	☐	☐	☐	☐	☐	☑	☐
4	Gaza is slaying in the darkness by Israel #starlinkforgaza https://t.co/ymrjCICRGw	The video shows an Algerian fireworks celebration and has	☐	☐	☐	☐	☐	☑	☐	☐	☐	☐	☑	☐
5	Caitlin Clark motion alert https://t.co/UQjLtgbR4N	This is digitally altered. The user claims it is real. Full live	☐	☐	☐	☑	☐	☐	☐	☐	☐	☐	☑	☐
6	Russia is conducting naval exercises using nuclear submarines just 1	This video is from 2018 and shows a missile launch in the	☐	☐	☐	☑	☐	☐	☐	☐	☐	☐	☑	☐
7	I had to wait on posting this because I wanted to make sure it was true Of course to suggest this money came from US and western tax payer If you listen to Zelensky speak you'd think he cares deeply about Ukra If this post gets a community note immediately vote it unhelpful. They a #UkraineRussianWar #ukraine #russia #Zelensky #putin #nato #BRIC	Missing VAT, currency, legal requirements for French invoic	☑	☑	☑	☐	☐	☑	☐	☐	☐	☐	☑	☑
8	18 million ballots were rejected and purged HOW?	They are not rejected, they are still being counted. Californ	☑	☑	☐	☐	☐	☐	☐	☐	☐	☐	☑	☐

Figure 8: Our annotation setup.

Multilingual Gloss-free Sign Language Translation: Towards Building a Sign Language Foundation Model

Sihan Tan[1,2], Taro Miyazaki[2], Kazuhiro Nakadai[1]
[1]Institute of Science Tokyo, [2]NHK Science & Technology Research Laboratories
{tansihan, nakadai} @ra.sc.e.titech.ac.jp
miyazaki.t-jw@nhk.or.jp

Abstract

Sign Language Translation (SLT) aims to convert sign language (SL) videos into spoken language text, thereby bridging the communication gap between the sign and the spoken community. While most existing works focus on translating a single sign language into a single spoken language (one-to-one SLT), leveraging multilingual resources could mitigate low-resource issues and enhance accessibility. However, multilingual SLT (MLSLT) remains unexplored due to language conflicts and alignment difficulties across SLs and spoken languages. To address these challenges, we propose a multilingual gloss-free model with dual CTC objectives for token-level SL identification and spoken text generation. Our model supports 10 SLs and handles one-to-one, many-to-one, and many-to-many SLT tasks, achieving competitive performance compared to state-of-the-art methods on three widely adopted benchmarks: multilingual SP-10, PHOENIX14T, and CSL-Daily.[1]

1 Introduction

Sign language translation (SLT) is a sophisticated cross-modal task that converts sign language (SL) into spoken language, serving as a crucial bridge between the deaf and hard-of-hearing community and the hearing world. Recent advancements in deep learning have significantly improved SLT performance, particularly through either *gloss-based* or *gloss-free*[2] approaches (Camgoz et al., 2018; Chen et al., 2022). While gloss-based methods benefit from intermediate linguistic supervision, they suffer from an information bottleneck, limiting their real-world applicability (Müller et al., 2023). In contrast, gloss-free methods directly learn from raw SL videos, making them more practical yet challenging. Despite progress in SLT, existing research predominantly focuses on translating a single SL into a single spoken language (*one-to-one* SLT). However, collecting large-scale annotated SL datasets is difficult. Leveraging multilingual resources could mitigate low-resource issues and enhance accessibility. Existing multilingual SLT (MLSLT) studies (Yin et al., 2022; Zhang et al., 2025) are mostly limited to specific datasets (*e.g.,* SP-10) or restricted to *many-to-one* translation. While MLSLT holds great potential, it often suffers from performance degradation due to **language conflicts**. For instance, we observed a BLEU drop of 1.50 in our universal training setting (In § 5 many-to-one). In addition, **alignment challenges** between SLs and spoken languages hinder the development of MLSLT. To address these limitations, we propose a multilingual gloss-free SLT model with token-level sign language identification (SLI), capable of handling diverse multilingual SLT scenarios. Our contributions are as follows:

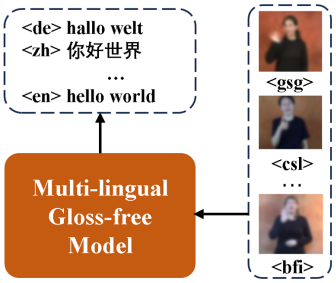


Figure 1: Overview of multilingual gloss-free model. Here, gsg = German Sign Language, csl = Chinese Sign Language, and bfi = British Sign Language.[3]

- We introduce *Sign2(LID+Text)*, a novel SLT approach that adopts dual CTC alignments: one with token-level SL IDs and the other with spoken languages, addressing the language conflicts and alignment challenges in MLSLT.

[1]Codes and model are available:https://github.com/Claire874/Gloss-free-MLSLT.

[2]Gloss is another written representation of sign language to help localize sign motions and simplify SLT tasks.

[3]We use the ISO639-3 and the ISO693-1 standard to represent sign language and spoken language.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 553–561
July 27 - August 1, 2025

- To the best of our knowledge, this is the first study to comprehensively explore one-to-one, many-to-one, and many-to-many gloss-free MLSLT, using multiple datasets (SP-10, PHOENIX14T, and CSL-Daily) and achieve state-of-the-art performance for each task.

2 Related work

Previous SLT studies mainly take a *cascading* or *end-to-end* approach. Cascading SLT, such as Sign2Gloss2Text, introduces gloss as intermediate supervision, simplifying SLT into two stages: sign language recognition (Sign2Gloss) and gloss-to-text translation (Gloss2Text) (Yin and Read, 2020). In contrast, end-to-end SLT directly converts sign videos into spoken texts. Camgoz et al. (2018) first proposed Sign2Text; however, it underperformed cascading SLT due to the challenging sign-text alignment. Later, Sign2Text was integrated with multi-task learning into Sign2(Gloss+Text) to alleviate the alignment issue (Camgoz et al., 2020). Recent work further advanced gloss-free SLT. Hamidullah et al. (2024) improved performance by introducing sentence embeddings as supervisions. In addition, large language models (LLMs) opened a new path for gloss-free SLT (Wong et al., 2024; Gong et al., 2024; Chen et al., 2024), but their applicability to multilingual settings is limited.

What is the current status of MLSLT? MLSLT remains underexplored due to several challenges. First, SLT itself involves the complex alignment between SL and spoken text, as SLs rely on fine-grained articulations such as finger spelling, palm orientation, and non-manual features (Liddell and Johnson, 1989). Second, language conflicts arise when training a unified model across diverse SLs. While certain SLs exhibit similarities, others differ greatly in structure and lexicon.[4] (Wei and Chen, 2023; Zhang et al., 2025). An intuitive solution is to introduce utterance-level SLI (Gebre et al., 2013); however, it is an ill-defined task, as models can learn to identify signers for particular SLs (Jiang et al., 2024). Inspired by token-level language identification in multilingual automatic speech recognition (ASR) (Chen et al., 2023), token-level SLI could offer a more flexible solution. Beyond resolving language conflicts, it could aid the model in mapping SLs to a large multilingual text space by providing fine-grained language cues throughout the sequence. Lastly, MLSLT suffers from a lack of large-scale datasets, and despite recent efforts (Yin et al., 2022; Gueuwou et al., 2023; Tanzer, 2024), data resources remain scarce.

[4]Examples are provided in Appendix A.

3 Method

To address the language conflict and alignment difficulties in MLSLT, we propose *Sign2(LID+Text)*, a novel approach that predicts token-level SL IDs (LID_{tok}) and translates SLs into spoken languages. Unlike previous studies (Gebre et al., 2013; Jiang et al., 2024) which predict an utterance-level LID (*e.g.,* a single label <ase> and <en> for American sign language), we introduce two auxiliary CTC objectives (Graves et al., 2006) to explicitly supervise LID_{tok} and target spoken text alignment, enabling hierarchical encoding under a joint CTC/Attention framework, as illustrated in Figure 2. This allows the early encoder layers to focus on token-level SLI, while the later layers reorder the latent sign representations for translation using a text-oriented CTC objective (TxtCTC). Table 1 summarizes the defined tasks and labels.

Tasks	Labels
MLSLT (many-to-many)	<en> hello world
MLSLT (many-to-one)	hello world
SLT (one-to-one)	hello world
Token-level SLI	<ase> <ase> <ase>

Table 1: Training label examples of Sign2 (LID+Text). In token-level SLI, all tokens are replaced with LIDs, while utterance-level SLI contains a single LID label.

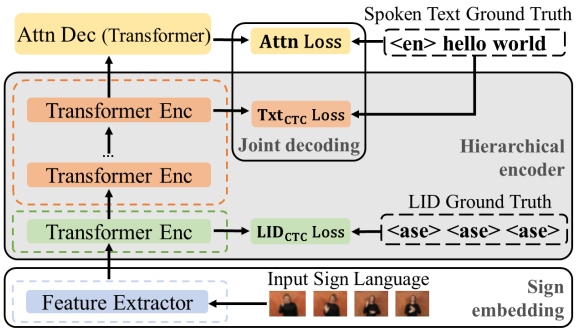


Figure 2: Overview of multilingual gloss-free model.

3.1 Feature extractor

In previous studies (Zhang et al., 2023; Tan et al., 2025), a pre-trained feature extractor on glosses has been used as sign embeddings, which is inherently designed for gloss-based SLT. Since our approach is gloss-free, we instead adopt a pre-trained feature extractor based on the SlowFastSign network (Ahn

et al., 2024) but trained on spoken texts. Pre-trained sign embeddings are further used to extract the sign feature $\mathcal{F}$ from the sign video sequence $\mathcal{V} = \{v_1, v_2, ..., v_{|\mathcal{V}|}\}$ consisting of $|\mathcal{V}|$ frames. This process is formulated as:

$$\mathcal{F} = \text{SignEmbedding}(\mathcal{V}), \tag{1}$$

where $\mathcal{F} = \{f_1, f_2, \ldots, f_{|\mathcal{F}|}\}$ denotes the extracted feature with $|\mathcal{F}|$ sign representations.

3.2 Hierarchical encoder

We employ a CTC-based hierarchical encoder, widely used in ASR (Sanabria and Metze, 2018) and machine translation (Yan et al., 2023), to facilitate multi-task learning and improve cross-modal alignments in MLSLT. The hierarchical encoder consists of two modules: an initial token-level SLI (Sign2LID) module and a subsequent Sign2Text module that reorders multilingual sign representations within a joint CTC/Attention framework, optimized with separate CTC objectives.

Sign2LID module is to predict the LID_{tok}, $\mathcal{I}_{\text{tok}} = \{i_1, i_2, ..., i_{|\mathcal{T}|}\}$ with the same length as the spoken text $\mathcal{T} = \{t_1, t_2, ..., t_{|\mathcal{T}|}\}$. We incorporate the LID_{tok}-oriented CTC loss as part of the multi-task Sign2(LID+Text) objective function.

$$\mathcal{L}_{\text{LID}} = -\log P_{\text{CTC}}(\mathcal{I}_{\text{tok}}|\mathcal{F}). \tag{2}$$

As in hierarchical conditioning, deeper layers handle increasingly complex predictions (Higuchi et al., 2022); the initial encoder layer suffices for the Sign2LID task. We assign this auxiliary task to the initial encoder layer, where the $|\mathcal{T}|$-length LID_{tok} sequence explicitly aligns the sign representations with each spoken word as

$$\mathbf{h}_{\text{int}} = \mathbf{Enc}_{\text{int}}(\mathcal{F}). \tag{3}$$

The intermediate sign representations $\mathbf{h}_{\text{int}}$ from the initial encoder layer, $\mathbf{Enc}_{\text{int}}$, are then forwarded to the subsequent encoder layers for Sign2Text.

Sign2Text module reorders the sign representations into the spoken text sequence. While CTC is typically constrained to monotonic alignments, neural network encoders allow for latent reordering (Zhang et al., 2022), enabling CTC to handle the non-monotonic alignment between the SL and spoken text. We apply TxtCTC to align the final encoder representation $\mathbf{h}_{\text{fin}}$ with the target spoken text sequence $\mathcal{T}$:

$$\mathcal{L}_{\text{Txt}} = -\log P_{\text{CTC}}(\mathcal{T}|\mathbf{h}_{\text{fin}}). \tag{4}$$

Following previous work (Yan et al., 2023), we frame our decoding process within a joint CTC/Attention setup, where the attention decoder plays a leading role in generating the output sequence, and the TxtCTC score provides auxiliary guidance during beam search. The overall training objective function jointly optimizes the hierarchical encoder and the attention decoder:

$$\mathcal{L}_{\text{total}} = \lambda_1\mathcal{L}_{\text{LID}} + \lambda_2\mathcal{L}_{\text{Txt}} + \lambda_3\mathcal{L}_{\text{Attn}}, \tag{5}$$

where $\mathcal{L}_{\text{Attn}}$ denotes the maximum likelihood estimation (MLE) loss for MLSLT, and λs control contributions of the Sign2LID, Sign2Text, and attention decoder objectives.

4 Experimental settings

To validate our proposed method, we conduct experiments on three tasks: one-to-one, many-to-one, and many-to-many SLT.

Datasets. We utilize three widely adopted datasets for our experiments: the multilingual SP-10 (Yin et al., 2022), RWTH-PHOENIX-2014T (PHOENIX14T) (Camgoz et al., 2018), and CSL-Daily (Zhou et al., 2021). SP-10 supports a broader range of tasks, featuring video recordings of 10 SLs from SpreadTheSign (Hilzensauer and Krammer, 2015). In contrast, PHOENIX14T and CSL-Daily are designed for one-to-one SLT. Appendix C provides statistics for the three datasets.

Implementation Details. We adopt a Transformer-based architecture within a joint CTC/Attention decoding framework (Tan et al., 2025). To evaluate the effectiveness of our method, we compare it with a vanilla Transformer baseline (Vaswani et al., 2017). Full implementation details and hyperparameters are provided in Appendix B.

Evaluation metrics. Following the previous studies, we evaluate performance using BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004). BLEU is calculated through SacreBLEU (Post, 2018).

5 Results and Discussion

One-to-one evaluates the alignment capability of TxtCTC in standard single-pair SLT. Since Sign2LID is not utilized in this task, it is a purely gloss-free SLT setting. Tables 2 and 3 present results on SP-10, PHOENIX14T, and CSL-Daily, respectively. Our TxtCTC, integrated within the

joint CTC/Attention framework, achieves 1.71 and 2.24 BLEU improvements on PHOENIX14T and CSL-Daily. To further investigate the effect of TxtCTC, we present the token length distribution of PHOENIX14T and CSL-Daily along with the average BLEU4 score on each interval (see Figures 3 and 4). We observed that proposed TxtCTC within the joint CTC/Attention framework tends to be more effective for short and medium-length sentences. The impact of TxtCTC diminishes as sentence length increases. CTC tends to be more effective for segments with fewer ambiguities and clearer frame-to-token correspondences, which are more common in shorter sequences. For longer sequences, the model relies more on the attention mechanism to capture global context, which may naturally reduce the marginal contribution of the TxtCTC objective. In addition, Gloss-free SLT introduces an additional challenge that the input SL frame sequence is typically much longer than the spoken sentence. This inherent length difference increases the alignment difficulty, particularly for long sentences.

As no prior work reports one-to-one SLT results on SP-10 beyond English, we provide the first benchmark to facilitate future research.

Language Pairs	Dev		Test	
	BLEU	ROUGE	BLEU	ROUGE
csl → zh	8.79	35.59	7.32	32.40
ukl → uk	7.47	32.56	6.84	30.12
rsl → ru	6.20	31.79	4.23	28.98
icl → is	4.79	27.57	4.25	30.45
gsg → de	6.07	33.73	5.77	32.20
ise → it	5.88	30.13	4.76	27.91
bqn→bg	4.77	28.93	2.59	23.91
swl → sv	7.47	31.31	7.23	30.45
lls → lt	2.33	26.70	2.42	24.36
bfi → en	7.80	33.77	6.23	32.33

Table 2: One-to-one SLT results on the SP-10 dataset.

Many-to-one evaluates many-to-one SLT on the SP-10 dataset. Following previous studies (Yin et al., 2022; Zhang et al., 2025), we selected English as the target spoken language. A major challenge in the many-to-one setting is the language conflict, as confirmed by a preliminary experiment: the baseline many-to-one model suffers an average BLEU drop of 1.50 compared to individually trained one-to-one models (see Table 4 individual(10) and universal (1)). Instead, our *Sign2(LID+Text)* approach mitigates the language conflict and surpasses individual translation by 0.58 BLEU. Table 4 reports the results of each SL. Overall, our model outperforms previous ML-SLT systems. However, the limited target vocabulary (∼1.1k words) constrains further improvements. Data augmentation could be a promising way to address this limitation.

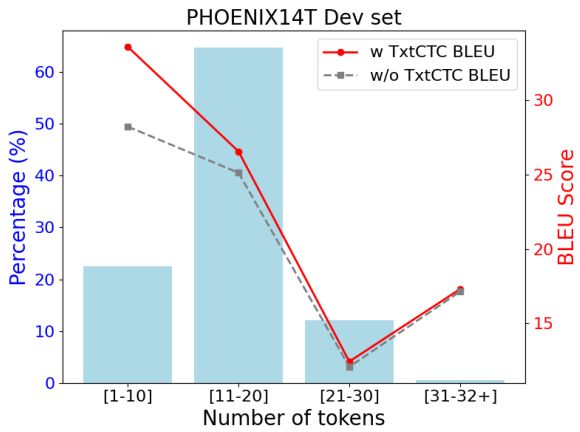


Figure 3: Average BLEU score on different token length intervals on PHOENIX14T.

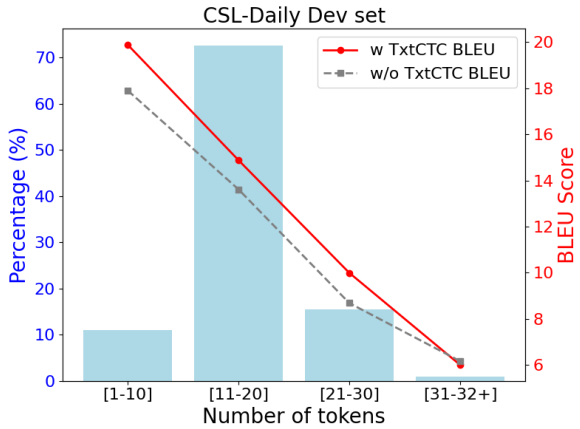


Figure 4: Average BLEU score on different token length intervals on CSL-Daily.

Many-to-many is the most challenging setting. We incrementally add language pairs based on their performance in Table 2, from highest to lowest on the dev set (see Appendix F). Table 5 presents the comparison with one-to-one SLT. Our many-to-many model maintains comparable performance as the number of language pairs increases to five. This stability benefits from cross-lingual information sharing, which reduces the reliance on large-scale data, particularly in low-resource SLT scenarios. To further validate LIDtok, we conduct an ablation study (see Appendix E) and find that LIDtok is especially effective under more challenging translation conditions. These results suggest that our many-to-many model provides a promising and scalable solution to mitigating the data scarcity

Methods	PHOENIX14T				CSL-Daily			
	Dev		Test		Dev		Test	
	BLEU	ROUGE	BLEU	ROUGE	BLEU	ROUGE	BLEU	ROUGE
Gloss-free								
NSLT+Luong (Camgoz et al., 2018)	10.00	32.60	9.00	30.70	7.96	34.28	7.56	34.54
CSGCR (Zhao et al., 2022)	15.08	38.96	15.18	38.85	–	–	–	–
GFSLT-VLP (Zhou et al., 2023)	22.12	43.72	21.44	42.49	11.07	36.70	11.00	36.44
Sign2GPT (Wong et al., 2024)	–	–	22.52	48.90	–	–	15.40	42.36
Fla-LLM (Chen et al., 2024)	–	–	23.09	45.27	–	–	14.20	37.25
SignLLM (Gong et al., 2024)	**25.25**	47.23	23.40	44.49	12.23	39.18	**15.75**	39.91
Baseline	22.59	49.88	22.52	49.85	12.23	36.39	11.76	36.25
Ours w TxtCTC	24.18	**51.74**	**24.23**	**50.60**	**13.66**	**39.33**	14.18	**40.00**

Table 3: Experimental results on PHOENIX14T and CSL-Daily dataset for gloss-free SLT (one-to-one SLT).

Part/Metrics	Methods	csl	ukl	rsl	bqn	icl	gsg	ise	swl	lls	bfi	Mean
Dev / BLEU	Individual (10)	**8.02**	6.32	5.56	**4.88**	5.03	5.54	4.78	7.54	5.15	6.42	5.92
	Universal (1)	5.93	5.32	4.91	3.15	4.69	4.65	4.52	6.63	4.47	4.59	4.89
	Google Multi (Johnson et al., 2017)	2.46	3.14	2.93	2.21	3.44	2.71	3.18	2.89	1.81	3.49	2.83
	MLST (Yin et al., 2022)	5.16	5.42	4.95	3.28	**6.76**	5.18	**7.05**	6.33	**6.08**	7.03	5.72
	Ours	7.06	**6.77**	**7.38**	3.56	6.59	**5.59**	4.83	**7.76**	4.54	**7.79**	**6.19**
Dev / ROUGE	Individual (10)	36.60	33.61	30.05	27.70	31.51	32.47	29.88	35.80	31.44	32.57	28.53
	Universal (1)	32.74	31.64	31.37	26.19	30.18	29.26	30.12	33.19	31.16	30.22	30.61
	Google multi (Johnson et al., 2017)	28.50	28.93	30.01	24.66	29.91	29.75	28.33	31.01	27.7	32.42	29.12
	MLSLT (Yin et al., 2022)	34.59	34.04	31.62	**27.98**	35.29	33.50	**37.96**	36.02	**34.48**	37.25	34.27
	Ours	**36.77**	**34.86**	**36.02**	26.74	**35.64**	**34.95**	31.67	**37.97**	31.43	**37.56**	**34.36**
Test / BLEU	Individual (10)	**6.24**	4.00	3.69	**3.63**	3.30	3.77	3.40	6.21	4.49	6.23	4.60
	Universal (1)	2.72	2.36	2.19	3.02	4.14	3.31	1.19	3.94	3.20	4.70	3.10
	Google Multi (Johnson et al., 2017)	2.28	2.38	2.06	1.10	1.38	1.82	2.09	2.13	2.68	3.27	2.12
	MLSLT (Yin et al., 2022)	5.19	4.18	3.66	2.85	3.93	**4.97**	**6.70**	3.70	**5.72**	5.73	4.66
	Ours	5.92	**4.52**	**5.80**	2.93	**5.10**	4.65	5.00	**6.40**	5.13	**6.36**	**5.18**
Test / ROUGE	Individual (10)	34.51	30.93	30.90	**27.19**	28.00	30.19	27.66	33.92	31.36	35.73	31.04
	Universal (1)	29.57	29.03	28.96	26.66	31.75	30.12	26.73	31.11	30.33	32.41	29.67
	Google multi (Johnson et al., 2017)	29.37	28.63	29.57	23.95	28.53	29.36	29.30	29.83	30.03	30.76	28.93
	MLSLT (Yin et al., 2022)	33.33	**34.07**	31.54	25.75	33.25	32.13	**35.37**	33.09	**33.11**	35.34	32.70
	Ours	**35.18**	33.17	**34.17**	24.55	**34.57**	**32.83**	34.27	**34.98**	31.48	**36.16**	**33.14**

Table 4: Many-to-one SLT results on the SP-10 dataset, we select English as the target spoken language. The best performance is bolded, and the second-best is underlined.

problem and paves the way toward a unified SL foundation model in challenging SLT settings.

Language Pairs	One-to-one		Many-to-many	
	BLEU	ROUGE	BLEU	ROUGE
(2→2)	6.28	32.37	6.22	34.53
(3→3)	6.44	31.73	6.15	34.02
(4→4)	6.41	34.59	6.31	32.76
(5→5)	6.33	33.88	5.36	31.79
(6→6)	6.08	33.40	4.91	31.17
(7→7)	5.97	34.37	4.63	29.51
(8→8)	5.84	32.70	4.75	30.24
(9→9)	5.27	32.14	4.74	28.65
(10→10)	5.06	32.31	4.58	30.83

Table 5: One-to-one vs. many-to-many SLT.

6 Conclusion

To address language conflicts and alignment challenges in multilingual sign language translation (MLSLT), we proposed *Sign2(LID+Text)*, a multilingual gloss-free SLT model combining token-level sign language identification (Sign2LID) and sign-to-text CTC alignment (Sign2Text). Our approach achieved comparable performance with the state-of-the-art across one-to-one, many-to-one, and many-to-many SLT tasks on three widely adopted benchmarks, covering a total of 10 different sign languages (SLs). We showed that Sign2LID effectively mitigates language conflicts and Sign2Text improves sign-to-text alignment, especially for shorter and medium-length sequences. Our work encourages and lays the foundation for future exploration of large-scale multilingual gloss-free SLT and shows potential for enhancing cross-lingual SL processing, contributing to the development of a universal SL foundation model.

Limitations

The limitations of this work can be summarized as follows. First, data scarcity remains a major challenge. The SP-10 dataset is currently the only publicly available multilingual SLT corpus, and

as shown in Appendix C, each language in SP-10 contains only 830 training samples (8.3k overall), which is extremely small for training deep learning models. Moreover, as discussed in the many-to-one setting, the target language vocabulary is limited to approximately 1.1k words, further constraining the model's capacity to generate diverse outputs. Second, our many-to-many SLT evaluation is set to, at most, 10 language pairs. Extending the evaluation to the full 10×10 combinations poses a greater challenge and requires more computational resources. Future work will focus on scaling to more sign languages and more challenging settings.

Acknowledgment

We thank the (meta-)reviewers for their valuable feedback.

References

Junseok Ahn, Youngjoon Jang, and Joon Son Chung. 2024. Slowfast network for continuous sign language recognition. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3920–3924.

Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 7784–7793.

Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020. Sign Language Transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10023–10033.

William Chen, Brian Yan, Jiatong Shi, Yifan Peng, Soumi Maiti, and Shinji Watanabe. 2023. Improving massively multilingual asr with auxiliary ctc objectives. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Yutong Chen, Ronglai Zuo, Fangyun Wei, Yu Wu, Shujie LIU, and Brian Mak. 2022. Two-stream network for sign language recognition and translation. In *Advances in Neural Information Processing Systems*, volume 35, pages 17043–17056. Curran Associates, Inc.

Zhigang Chen, Benjia Zhou, Jun Li, Jun Wan, Zhen Lei, Ning Jiang, Quan Lu, and Guoqing Zhao. 2024. Factorized learning assisted with large language model for gloss-free sign language translation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7071–7081, Torino, Italia. ELRA and ICCL.

Binyam Gebrekidan Gebre, Peter Wittenburg, and Tom Heskes. 2013. Automatic sign language identification. In *2013 IEEE International Conference on Image Processing*, pages 2626–2630.

Jia Gong, Lin Geng Foo, Yixuan He, Hossein Rahmani, and Jun Liu. 2024. Llms are good sign language translators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18362–18372.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *International Conference on Machine Learning*, page 369–376. Association for Computing Machinery.

Shester Gueuwou, Sophie Siake, Colin Leong, and Mathias Müller. 2023. JWSign: A highly multilingual corpus of Bible translations for more diversity in sign language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9907–9927, Singapore. Association for Computational Linguistics.

Yasser Hamidullah, Josef van Genabith, and Cristina España-Bonet. 2024. Sign language translation with sentence embedding supervision. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 425–434, Bangkok, Thailand. Association for Computational Linguistics.

Yosuke Higuchi, Keita Karube, Tetsuji Ogawa, and Tetsunori Kobayashi. 2022. Hierarchical conditional end-to-end asr with ctc and multi-granular subword units. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7797–7801. IEEE.

Marlene Hilzensauer and Klaudia Krammer. 2015. A multilingual dictionary for sign languages:" spreadthesign". In *ICERI2015 Proceedings*, pages 7826–7834. IATED.

Zifan Jiang, Gerard Sant, Amit Moryossef, Mathias Müller, Rico Sennrich, and Sarah Ebling. 2024. SignCLIP: Connecting text and sign language by contrastive learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9171–9193, Miami, Florida, USA. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Scott K. Liddell and Robert E. Johnson. 1989. American sign language: The phonological base. *Sign Language Studies*, (64):195–278.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Mathias Müller, Zifan Jiang, Amit Moryossef, Annette Rios, and Sarah Ebling. 2023. Considerations for meaningful sign language machine translation based on glosses. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 682–693, Toronto, Canada. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ramon Sanabria and Florian Metze. 2018. Hierarchical multitask learning with ctc. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 485–490.

Sihan Tan, Taro Miyazaki, Nabeela Khan, and Kazuhiro Nakadai. 2025. Improvement in sign language translation using text CTC alignment. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3255–3266, Abu Dhabi, UAE. Association for Computational Linguistics.

Garrett Tanzer. 2024. Fleurs-asl: Including american sign language in massively multilingual multitask evaluation. *Preprint*, arXiv:2408.13585.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Fangyun Wei and Yutong Chen. 2023. Improving continuous sign language recognition with cross-lingual signs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23612–23621.

Ryan Wong, Necati Cihan Camgoz, and Richard Bowden. 2024. Sign2GPT: Leveraging large language models for gloss-free sign language translation. In *The Twelfth International Conference on Learning Representations*.

Brian Yan, Siddharth Dalmia, Yosuke Higuchi, Graham Neubig, Florian Metze, Alan W Black, and Shinji Watanabe. 2023. CTC alignments improve autoregressive translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1623–1639, Dubrovnik, Croatia. Association for Computational Linguistics.

Aoxiong Yin, Zhou Zhao, Weike Jin, Meng Zhang, Xingshan Zeng, and Xiaofei He. 2022. Mlslt: Towards multilingual sign language translation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5099–5109.

Kayo Yin and Jesse Read. 2020. Better sign language translation with STMC-transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5975–5989, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Biao Zhang, Barry Haddow, and Rico Sennrich. 2022. Revisiting end-to-end speech-to-text translation from scratch. In *International Conference on Machine Learning*, pages 26193–26205. PMLR.

Biao Zhang, Mathias Müller, and Rico Sennrich. 2023. SLTUNET: A simple unified model for sign language translation. In *The Eleventh International Conference on Learning Representations*.

Ruiquan Zhang, Cong Hu, Pei Yu, and Yidong Chen. 2025. Improving multilingual sign language translation with automatically clustered language family information. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3579–3588, Abu Dhabi, UAE. Association for Computational Linguistics.

Jian Zhao, Weizhen Qi, Wengang Zhou, Nan Duan, Ming Zhou, and Houqiang Li. 2022. Conditional sentence generation and cross-modal reranking for sign language translation. *IEEE Transactions on Multimedia*, 24:2662–2672.

Benjia Zhou, Zhigang Chen, Albert Clapés, Jun Wan, Yanyan Liang, Sergio Escalera, Zhen Lei, and Du Zhang. 2023. Gloss-free sign language translation: Improving from visual-language pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20871–20881.

Hao Zhou, Wengang Zhou, Weizhen Qi, Junfu Pu, and Houqiang Li. 2021. Improving sign language translation with monolingual data by sign back-translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1316–1325.

A Examples of sign language conflict

As shown in Figure 5, many sign languages share similarities in expressing certain concepts. For example, when signing *rain*, signers often mimic the shape of raindrops falling, which is relatively universal. However, for *evening*, although the core concept involves representing the sun setting, variations in expression still exist across different sign languages.

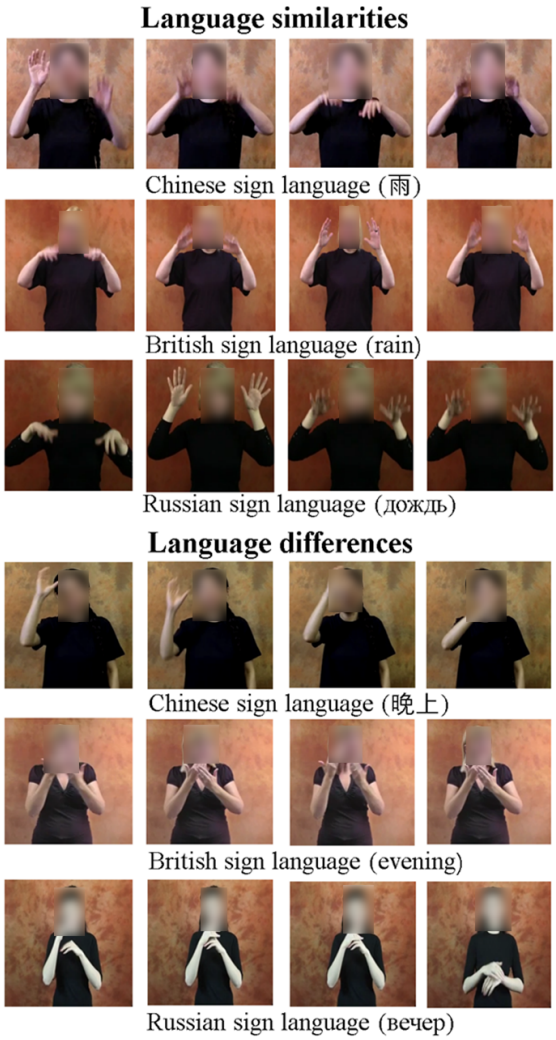


Figure 5: Sign language similarities and differences across languages. Sign videos are from SpreadTheSign. Note for privacy: we anonymize signers.

B Implement details

The Transformer model with a CTC/Attention setup uses a hidden size of 256 and a feed-forward dimension of 2048. Both the encoder and decoder have six layers. Training is conducted using the Xavier initializer and the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 1×10^{-3}. We train the model on an NVIDIA A100 (80GB) GPU with a batch size of 64. The total trainable #params is 39.48M. The training objective weights are set as $\lambda_1 = 1$, $\lambda_2 = 5$, and $\lambda_3 = 3$. When token-level LID is not used, λ_1 is set to 0.

C Summary of different SLT datasets

We summarize the statistics of the different datasets in Table 6, SP-10 consists of 10 different sign languages, with each having 830 training samples. In addition, Table 7 shows the involved sign languages with their abbreviations in SP-10.

Datasets	Lang	Statistics				
		#Signer	Vocab	#Train	#Dev	#Test
SP-10	multilingual	79	16.7k	8,300	1,420	2,021
PHOENIX14T	gsg	9	2.9k	7,096	519	642
CSL-Daily	csl	10	2.3k	18,401	1,077	1,176

Table 6: Statistics of SP-10, PHOENIX14T, and CSL-Daily datasets.

Languages	Abbr.
Chinese sign language	csl
Ukrainian sign language	ukl
Russian sign language	rsl
Icelandic sign language	icl
German sign language	gsg
Italian sign language	ise
Bulgarian sign language	bqn
Swedish sign language	swl
Lithuanian sign language	lls
British sign language	bfi

Table 7: Sign language abbreviations of the SP-10 dataset.

D The language conflict in SP-10

We performed a preliminary experiment to investigate language conflicts in multilingual SLT. The baseline model is adopted for individual (10) and universal (1) many-to-one SLT. The individual and universal translation performances are presented in Table 8. In general, the universal has an average 1.50 BLEU performance drop.

E Ablation study of token-level LID

Performance deteriorates as the number of languages involved in the many-to-many translation increases, and the translation task becomes more complex. As shown in Table 9, our method using token-level LID can suppress this deterioration and is effective in more complex translation settings.

Language Pairs	Individual (10) BLEU	Universal (1) BLEU	Variation
csl →en	6.24	2.72	−3.52
ukl → en	4.00	2.36	−1.64
rsl → en	3.69	2.19	−1.50
icl → en	3.30	4.14	+0.84
gsg → en	3.77	3.31	−0.46
ise → en	3.40	1.19	−2.21
bqn→en	3.63	3.02	−0.61
swl → en	6.21	3.94	−2.27
lls → en	4.49	3.20	−1.29
bfi → en	6.23	4.70	−1.53
Mean	**4.60**	**3.10**	**−1.50**

Table 8: Language conflicts in SP-10, we present the individual and universal translation results on the baseline.

Language Pairs	w/o LID_{tok} BLEU	w LID_{tok} BLEU
(2→2)	**7.48**	6.22
(3→3)	**6.50**	6.15
(4→4)	4.98	**6.31**
(5→5)	4.74	**5.36**
(6→6)	4.57	**4.91**
(7→7)	3.70	**4.63**
(8→8)	4.27	**4.75**
(9→9)	3.56	**4.74**
(10→10)	3.87	**4.58**

Table 9: Ablation study of token-level language ID (LID_{tok}) in many-to-many SLT.

F Details of language pairs in many-to-many SLT

Table 10 shows each language pair used in our many-to-many translation experiment.

Language Pairs	
(2→2)	(csl→zh) (bfi→en)
(3→3)	(csl→zh) (bfi→en) (swl→sv)
(4→4)	(csl→zh) (bfi→en) (swl→sv) (ukl→uk)
(5→5)	(csl→zh) (bfi→en) (swl→sv) (ukl→uk) (gsg→de)
(6→6)	(csl→zh) (bfi→en) (swl→sv) (ukl→uk) (gsg→de) (ise→it)
(7→7)	(csl→zh) (bfi→en) (swl→sv) (ukl→uk) (gsg→de) (ise→it) (rsl→ru)
(8→8)	(csl→zh) (bfi→en) (swl→sv) (ukl→uk) (gsg→de) (ise→it) (rsl→ru) (icl→is)
(9→9)	(csl→zh) (bfi→en) (swl→sv) (ukl→uk) (gsg→de) (ise→it) (rsl→ru) (icl→is) (bqn→bg)
(10→10)	(csl→zh) (bfi→en) (swl→sv) (ukl→uk) (gsg→de) (ise→it) (rsl→ru) (icl→is) (bqn→bg) (lls→lt)

Table 10: Language pairs used in many-to-many SLT

Advancing Sequential Numerical Prediction in Autoregressive Models

Xiang Fei[*1] **Jinghui Lu**[*1] **Qi Sun**[*2] **Hao Feng**[†1] **Yanjie Wang**[1]
Wei Shi[1] **An-lan Wang**[1] **Jingqun Tang**[1] **Can Huang**[†1]
[1]ByteDance Inc. [2]City University of Hong Kong
{feixiang.77, lujinghui, fenghao.2019}@bytedance.com
{wangyanjie.prince, shiwei.11, wanganlan}@bytedance.com
{tangjingqun, can.huang}@bytedance.com
qisun.new@gmail.com

Abstract

Autoregressive models have become the de facto choice for sequence generation tasks, but standard approaches treat digits as independent tokens and apply cross-entropy loss, overlooking the coherent structure of numerical sequences. This paper introduces ***N**umerical **T**oken **I**ntegrity **L**oss (NTIL)* to address this gap. NTIL operates at two levels: (1) token-level, where it extends the Earth Mover's Distance (EMD) to preserve ordinal relationships between numerical values, and (2) sequence-level, where it penalizes the overall discrepancy between the predicted and actual sequences. This dual approach improves numerical prediction and integrates effectively with LLMs/MLLMs. Extensive experiments show significant performance improvements with NTIL. All resources are available at https://github.com/xfey/NTIL.

1 Introduction

In recent years, sequence generation has become a crucial approach for implementing a broad range of AI applications, including visual question answering (Wang et al., 2024d; Reich and Schultz, 2024; Fan et al., 2024; Liu et al., 2024b; He et al., 2025; Wang et al., 2025; Lu et al., 2024b), key information extraction (Kim et al., 2024; Yu et al., 2024; Kang et al., 2024; Wang et al., 2024a; Yi et al., 2025; Lu et al., 2023a, 2024a), object detection (Wen et al., 2024), math reasoning (Zhao et al., 2024), text spotting (Li et al., 2024; Yu et al., 2025), and automatic audio recognition (Zhou et al., 2024).

Autoregressive models, especially large language models (LLMs) such as GPT (Achiam et al., 2023), LLaMA (Touvron et al., 2023; Dubey et al., 2024), Qwen (Yang et al., 2024; Wang et al., 2024c) series, with multi-modal large language models (MLLMs) based on them, now dominate the sequence generation tasks. During training, these models generate sequences token-by-token, typically using cross-entropy (CE) loss, to minimize the negative log-likelihood of the ground truth token at each time step. However, CE loss has several inherent limitations when predicting numerical values. Specifically, CE suffers from **Limitation 1** that *it ignores the inherent closeness between numerical tokens, where each digit in a numerical prediction is not independent but related to its neighboring digits.* For example, in Figure 2(a) and 2(b), for the ground truth token "3", the CE loss yields same values of $-log(0.5)$ for different prediction distributions. However, the distribution in Figure 2(b) is more accurate, as it assigns higher probability to the neighboring token "2".

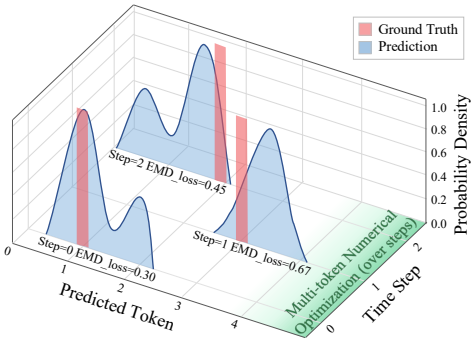


Figure 1: Sequence-level digit token loss illustration.

CE also suffers from **Limitation 2** that *it fails to capture the holistic numerical error when sequential tokens are involved, as it focuses on the precision of each token rather than the overall value*. In an autoregressive generation manner, producing a numerical value typically requires consecutive time steps. For example, the target value *"0.98"* requires the prediction of four sequential tokens — *"0"*, *"."*, *"9"*, *"8"*. Thus, a prediction such as *1.01* (*"1"*,*"."*,*"0"*,*"1"*) incurs a high CE loss as the first, third and fourth tokens are significantly differ-

[*]Equal Contribution
†Corresponding Author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 562–574
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

ent from the target tokens. Conversely, a prediction like *1.98* (*"1"*,*"."*,*"9"*,*"8"*) could yield a lower CE loss due to a closer match at the token level, despite the overall numerical difference being larger (1.00 vs. 0.03). This discrepancy shows the limitation of CE in evaluating predictions holistically.

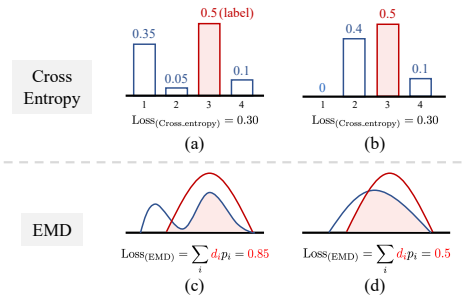


Figure 2: Cross-entropy fails to distinguish predictions, whereas EMD correlates smaller loss for better predicted distributions.

To overcome the above issues, we introduce a novel sequence-level numerical prediction loss: ***N**umerical **T**oken **I**ntegrity **L**oss (NTIL)*. At the token-level, NTIL replaces the traditional CE loss with Earth Mover's Distance (EMD) (Rubner et al., 1998). Additionally, we enhance the EMD with an *Exponential Position-based Weighting* scheme (Section 3.1), which leverages place-value number systems to better capture the nuanced differences between numerical distributions at each time step. At the sequence-level, NTIL evaluates the overall numerical difference between predicted and actual sequences through *Multi-token Numerical Optimization* (Section 3.2), considering all time steps holistically, as illustrated in Figure 1. It enables NTIL to effectively model the actual value of digit sequences, and capture discrepancies across the consecutive numerical range, moving beyond simple token-by-token comparison.

To the best of the authors' knowledge, it is the first time that EMD is used as an optimization method for autoregressive models. Moreover, our holistic approach is the first of its kind to improve sequential numerical prediction by considering numerical tokens across multiple time steps. Our method can be seamlessly integrated into both LLMs and MLLMs. Experimental results show that NTIL boosts performance in tasks requiring precise numerical outputs, such as object detection, text spotting, and math reasoning (Section 4).

2 Related Work

The Earth Mover's Distance (EMD) measures the minimal cost of transforming one distribution into another, and has become a valuable metric in deep learning applications. Notably, Wasserstein GAN (Arjovsky et al., 2017) uses EMD as its loss function to stabilize training in GANs. Cuturi (2013) and Courty et al. (2016) also adopted EMD for smoothing the training procedure. Despite the success of EMD, it has not been applied to autoregressive models. Most recently, autoregressive models, especially LLMs, have advanced NLP (Radford et al., 2019; Touvron et al., 2023; Lu et al., 2023b; Cui et al., 2025), and multi-modal tasks (Alayrac et al., 2022; Wang et al., 2024c; Feng et al., 2025; Lu et al., 2025). While the tasks mentioned above require high precision in numerical value prediction, none of the previous works have specifically optimized for this criterion. Our work addresses this gap by focusing on advancing the sequential numerical prediction for autoregressive models.

3 Method

This section details the components of the proposed method. Section 3.1 proposes exponential weighted EMD to single digits; Section 3.2 describes how we go through multiple digital tokens to derive a simple yet effective numerical measure.

3.1 Exponential Position-Based Weighting

For token-level prediction, to address **Limitation 1** in Section 1, we replace the conventional CE loss with EMD to account for the ordinal relationship during optimization. The preliminaries for both CE and EMD objectives, and the simplification via numerical prediction, are outlined in Appendix D.

Furthermore, we extend EMD to account for the place-value number systems, where leading digits have greater numerical significance. We implement an exponential weighting scheme to progressively assign weights based on digit positions, to scale their contributions to the loss accordingly:

$$\mathbf{W}_{\mathbf{exp}} = \left[(1+\sigma)^{n-i-1}\right]_{i=0}^{n-1}, \qquad (1)$$

where σ is the exponential increment rate, and n is the length of consecutive digits. This implementation helps the model understand the order relationship between consecutive numbers.

3.2 Multi-Token Numerical Optimization

To overcome **Limitation 2** outlined in Section 1, we propose the following procedure and losses. **Differentiable Numerical Value Construction.** In this step, we construct the complete numerical value from consecutive discrete digital tokens. Figure 3 illustrates how we obtain the digit index from the predicted distribution using argmax to derive the integer representation. To maintain differentiability, we employ the Gumbel-softmax approximation with reduced temperature and noise parameters to ensure consistent results. The resulting tensor is element-wise multiplied with positional indices, scaled by the appropriate powers of 10, and aggregated to obtain the final value. For further implementation details, see Appendix C.

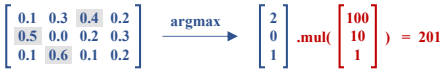


Figure 3: Constructing a numerical value from tokens.

Relative Deviation Metric. For numerical comparison, while absolute difference provides a straightforward measure equivalent to L1 loss, we propose a normalized metric defined as:

$$\mathcal{L}_{relative} = \frac{|X - Y|}{\max(X, Y) + \epsilon}, \tag{2}$$

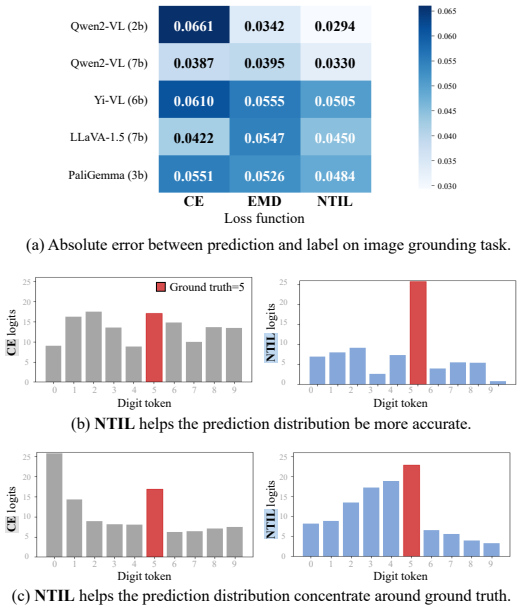
where X is the sequence-level numerical prediction (*e.g.*, "234") and Y is the ground truth, and ϵ is a small quantity to avoid division by zero. This normalization ensures consistency across different magnitude ranges.

Magnitude Deviation Metric We also apply a normalized metric on the order of magnitude as:

$$\mathcal{L}_{magnitude} = \log\left(\frac{\max(X, Y)}{\min(X, Y)}\right). \tag{3}$$

The objective penalizes the difference in the order of magnitude between two values. For example, given the pairs $(1, 10)$ and $(1, 100)$, which have similar $\mathcal{L}_{relatvie}$ values 0.90 and 0.99, but differ in $\mathcal{L}_{magnitude}$ value: $\log\left(\frac{10}{1}\right) \approx 2.30$ for the first pair and $\log\left(\frac{100}{1}\right) \approx 4.61$ for the second. This results in a larger penalty for greater differences in magnitude. The final formulation of NTIL combines the above loss functions, with tunable hyperparameters to weight their individual contributions.

$$\mathcal{L} = \mathbf{W}_{\mathbf{exp}}\,\mathrm{EMD} + \alpha \cdot \mathcal{L}_{relative} + \beta \cdot \mathcal{L}_{magnitude} \tag{4}$$



(a) Absolute error between prediction and label on image grounding task.

(b) **NTIL** helps the prediction distribution be more accurate.

(c) **NTIL** helps the prediction distribution concentrate around ground truth.

Figure 4: Results for quantitative analysis.

4 Experiments and Results

This section presents a comprehensive empirical evaluation of the proposed NTIL across various LLMs/MLLMs (Section 4.1). **CE** (Shannon, 1948) and **EMD** (Rubner et al., 1998) are chosen as baselines due to their widespread adoption. The evaluation encompasses multiple task domains that focus on numerical prediction including *Image Grounding*, *Scene Text Detection*, *Clock Time Recognition*, *Mathematical Reasoning* and *Arithmetic Calculations*. Appendix B provides details on tasks, datasets, and evaluation metrics. We also conduct systematic ablation studies to evaluate the critical components of our approach. Implementation details are available in Appendix A.

4.1 Main Results

4.2 Results of MLLMs

Image Grounding As shown in Table 1, our method outperforms both CE and EMD across nearly all datasets and VLM backbones, as evidenced by the overall performance improvements.

Scene Text Detection Table 2 shows that our method improves accuracy across multiple datasets, demonstrating its effectiveness in predicting multiple object coordinates.

Clock Time Recognition Table 4 demonstrates that NTIL surpasses CE and EMD significantly in performance across all model architectures.

Mathematical Reasoning As shown in Table 5, our method outperforms CE and EMD across all

Model	Method	RefCOCO			RefCOCO+			RefCOCOg		Avg
		Val	TestA	TestB	Val	TestA	TestB	Val	Test	
PaliGemma (3b) (Beyer et al., 2024)	CE	0.839	0.865	0.784	0.740	0.797	0.664	0.792	0.797	0.785
	EMD	0.841	0.864	**0.796**	0.749	0.805	0.669	0.789	0.799	0.789
	Ours	**0.844**	**0.873**	0.791	**0.750**	**0.812**	**0.678**	**0.804**	**0.802**	**0.795**
LLaVA-1.5 (7b) (Liu et al., 2024a)	CE	0.855	0.880	0.813	**0.801**	0.843	0.741	0.799	0.816	0.818
	EMD	0.856	0.879	**0.822**	0.798	0.845	0.743	0.798	0.816	0.820
	Ours	**0.858**	**0.885**	0.815	0.800	**0.853**	**0.747**	**0.802**	**0.817**	**0.822**
Yi-VL (6b) (Young et al., 2024)	CE	0.767	0.796	0.734	0.706	0.757	0.651	0.722	0.731	0.733
	EMD	**0.779**	0.805	0.738	**0.719**	0.762	0.657	0.721	0.737	0.740
	Ours	0.777	**0.808**	**0.741**	0.717	**0.770**	**0.665**	**0.727**	**0.743**	**0.744**
Qwen2-VL (2b) (Wang et al., 2024c)	CE	0.897	0.928	**0.850**	0.841	**0.896**	0.776	0.851	**0.867**	0.863
	EMD	0.889	0.931	0.843	0.838	0.889	0.772	0.853	0.858	0.859
	Ours	**0.898**	**0.932**	0.849	**0.844**	0.891	**0.788**	**0.858**	0.863	**0.866**
Qwen2-VL (7b) (Wang et al., 2024c)	CE	**0.892**	0.929	0.841	0.842	0.902	0.784	0.843	0.848	0.860
	EMD	0.886	0.926	0.834	0.843	0.901	0.768	0.836	0.843	0.855
	Ours	0.889	**0.931**	**0.840**	**0.844**	**0.904**	**0.786**	**0.848**	**0.853**	**0.862**

Table 1: Performance comparison (Acc@0.5) of models on image grounding tasks.

Model	Method	Dataset				Avg
		CTW1500	ICDAR1500	TD500	Total-Text	
PaliGemma (3b)	CE	0.220	0.129	0.183	0.259	0.193
	EMD	0.314	0.124	0.252	0.307	0.241
	Ours	**0.369**	**0.155**	**0.257**	**0.318**	**0.263**
Yi-VL (6b)	CE	**0.682**	0.370	0.753	0.673	0.586
	EMD	0.668	0.398	**0.778**	0.678	0.594
	Ours	0.680	**0.403**	0.752	**0.678**	**0.597**
Qwen2-VL (2b)	CE	0.786	0.538	0.851	0.827	0.720
	EMD	**0.786**	0.535	**0.867**	0.808	0.718
	Ours	0.776	**0.577**	0.854	**0.835**	**0.732**
Qwen2-VL (7b)	CE	**0.771**	0.648	**0.889**	0.864	0.764
	EMD	0.762	0.625	0.874	0.860	0.751
	Ours	0.770	**0.669**	0.869	**0.872**	**0.770**
LLaVA-1.5 (7b)	CE	0.735	0.490	0.821	0.786	0.675
	EMD	0.724	0.545	**0.840**	0.776	0.690
	Ours	**0.739**	**0.547**	0.839	**0.791**	**0.698**

Table 2: Performance (Acc@0.5) on scene text detection tasks.

Model	Accuracy (%)		
	CE	EMD	**Ours**
Baichuan2 (7b) (Yang et al., 2023)	44.3	46.6	**46.9**
Qwen2.5 (1.5b) (Team, 2024)	40.3	40.7	**42.4**
LLaMA3 (8b) (Dubey et al., 2024)	61.9	61.8	**61.9**
Yi (6b) (Young et al., 2024)	53.0	**54.6**	54.4
MiniCPM3 (4b) (Hu et al., 2024)	66.8	68.2	**68.6**

Table 3: Performance comparison of accuracies on the arithmetic calculation task.

Metric	Method	LLaVA-1.5 (7b)	Qwen2-VL (7b)	Qwen2-VL (2b)	Yi-VL (6b)
Accuracy (%) ↑	CE	95.1	75.0	81.3	76.2
	EMD	95.3	78.7	81.7	75.1
	Ours	**98.3**	**80.5**	**85.3**	**87.4**
Time gap (minute) ↓	CE	8.52	30.84	32.34	56.58
	EMD	7.98	30.78	31.98	54.78
	Ours	**4.14**	**27.72**	**24.66**	**26.58**

Table 4: Performance of the clock time recognition task.

Dataset	Method	Qwen2-vl (2b)	Qwen2-vl (7b)	LLaVA-1.5 (7b)	Yi-VL (6b)	PaliGemma (3b)
Mathvision	CE	0.139	0.184	0.146	0.143	0.097
	EMD	0.130	0.188	**0.148**	0.142	0.088
	Ours	**0.145**	**0.191**	0.146	**0.153**	**0.098**
Mathvista	CE	0.248	**0.315**	0.140	0.187	0.143
	EMD	**0.262**	0.303	0.157	0.192	0.149
	Ours	0.251	0.300	**0.170**	**0.222**	**0.157**

Table 5: Performance of the math reasoning task.

datasets, with the most significant improvements seen in the Mathvision dataset using the Qwen2-VL (2b) and in Mathvista with the Yi-VL (6b).

4.3 Results of LLMs

Arithmetic Calculation As shown in Table 3, our method improves accuracy across multiple LLMs, though LLaMA3 shows minimal gains, possibly due to its extensive pre-training. Overall, the majority of cases show that for numerical predictions, while EMD performs comparably or marginally better than CE loss, NTIL consistently delivers superior results in most scenarios. This underscores the effectiveness and generalizability of NTIL.

			PaliGemma		LLaVA-1.5		Qwen2-VL	Yi-VL
Exp	Rel	Mag	Mathvision	Mathvista	Mathvision	Mathvista	Clock_Time	Clock_Time
×	✓	✓	0.096	0.137	0.151	0.166	0.798	0.834
✓	×	✓	0.095	0.137	0.145	0.154	0.790	0.856
✓	✓	×	0.094	0.142	**0.160**	0.143	0.816	**0.876**
✓	✓	✓	**0.098**	**0.157**	0.146	**0.170**	**0.853**	0.874

Table 6: Ablations on NTIL. Exp: Exponential Position-Based Weighting. REL: Relative Deviation Metric. Mag: Magnitude Deviation Metric.

4.4 Ablation Analysis

Table 6 indicates that incorporating all components of NTIL generally leads to better performance, as evidenced by the highest scores in most metrics when all components are enabled. As an exception, the inclusion of Magnitude leads to worse results in Mathvision for LLaVA-1.5, which indicates the fluctuation of applying Magnitude in some cases.

4.5 Quantitative Analysis

As shown in Figure 4(a), NTIL achieves the lowest absolute errors among all models, indicating more consistent performance compared to CE and EMD. Figure 4(b) and 4(c) illustrate that NTIL pro-

duces more accurate predictions with distributions more concentrated around the ground truth. Overall, NITL offers more stability and lower variability. Qualitative examples can be seen in Appendix F.

5 Conclusion

We propose NTIL, which improves numerical prediction accuracy in LLMs at both the token and sequence levels. Experiments show improvement across multiple datasets and models, highlighting effectiveness of NTIL.

Limitations

The limitations of the NTIL include the exponential position-based weighting scheme, while effective in many cases, had limited or negative impact in certain configurations, such as with the Mathvision dataset and LLaVA-1.5 model. Future exploration could focus on refining the exponential position-based weighting scheme with adaptive strategies to address its inconsistent impact.

Furthermore, NTIL introduces additional computational overhead compared to CE loss, resulting in reduced training efficiency. This trade-off between performance improvement and computational cost needs to be considered in practical applications.

Another limitation arises from the tokenization strategies of certain models like LLaMA-3, which encode common multi-digit numbers (e.g., "123") as a single token. Such cases require special handling in NTIL's implementation, adding complexity to the framework.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Martin Arjovsky, S Chintala, and Léon Bottou. 2017. Wasserstein gan. arxiv preprint arxiv: 170107875. *arXiv preprint arXiv:1701.07875*.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. 2024. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*.

Chee Kheng Ch'ng and Chee Seng Chan. 2017. Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 935–942. IEEE.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. 2016. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865.

Xiao Cui, Mo Zhu, Yulei Qin, Liang Xie, Wengang Zhou, and Houqiang Li. 2025. Multi-level optimal transport for universal cross-tokenizer knowledge distillation on language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23724–23732.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yue Fan, Jing Gu, Kaiwen Zhou, Qianqi Yan, Shan Jiang, Ching-Chen Kuo, Yang Zhao, Xinze Guan, and Xin Wang. 2024. Muffin or Chihuahua? challenging multimodal large language models with multipanel VQA. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6845–6863, Bangkok, Thailand. Association for Computational Linguistics.

Hao Feng, Shu Wei, Xiang Fei, Wei Shi, Yingdong Han, Lei Liao, Jinghui Lu, Binghong Wu, Qi Liu, Chunhui Lin, et al. 2025. Dolphin: Document image parsing via heterogeneous anchor prompting. *arXiv preprint arXiv:2505.14059*.

gpiosenka. 2022. Time-image dataset-classification.

Yangfan He, Jianhui Wang, Kun Li, Yijin Wang, Li Sun, Jun Yin, Miao Zhang, and Xueqian Wang. 2025. Enhancing intent understanding for ambiguous prompts through human-machine co-adaptation. *arXiv preprint arXiv:2501.15167*.

Le Hou, Chen-Ping Yu, and Dimitris Samaras. 2016. Squared earth mover's distance-based loss for training deep neural networks. *arXiv preprint arXiv:1611.05916*.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

Iris AM Huijben, Wouter Kool, Max B Paulus, and Ruud JG Van Sloun. 2022. A review of the gumbel-max trick and its extensions for discrete stochasticity in machine learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(2):1353–1371.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Hyeonseok Kang, Hyein Seo, Jeesu Jung, Sangkeun Jung, Du-Seong Chang, and Riwoo Chung. 2024. Guidance-based prompt data augmentation in specialized domains for named entity recognition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 665–672, Bangkok, Thailand. Association for Computational Linguistics.

Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. 2015. Icdar 2015 competition on robust reading. In *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 1156–1160. IEEE.

Seoyeon Kim, Kwangwook Seo, Hyungjoo Chae, Jinyoung Yeo, and Dongha Lee. 2024. VerifiNER: Verification-augmented NER via knowledge-grounded reasoning with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2441–2461, Bangkok, Thailand. Association for Computational Linguistics.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Wenhao Liu, Xiaohua Wang, Muling Wu, Tianlong Li, Changze Lv, Zixuan Ling, Zhu JianHao, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. 2024b. Aligning large language models with human preferences through representation engineering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10619–10638, Bangkok, Thailand. Association for Computational Linguistics.

Jinghui Lu, Yanjie Wang, Ziwei Yang, Xuejing Liu, Brian Mac Namee, and Can Huang. 2024a. Padellmner: parallel decoding in large language models for named entity recognition. *Advances in Neural Information Processing Systems*, 37:117853–117880.

Jinghui Lu, Haiyang Yu, Yanjie Wang, Yongjie Ye, Jingqun Tang, Ziwei Yang, Binghong Wu, Qi Liu, Hao Feng, Han Wang, et al. 2024b. A bounding box is worth one token: Interleaving layout and text in a large language model for document understanding. *arXiv preprint arXiv:2407.01976*.

Jinghui Lu, Haiyang Yu, Siliang Xu, Shiwei Ran, Guozhi Tang, Siqi Wang, Bin Shan, Teng Fu, Hao Feng, Jingqun Tang, et al. 2025. Prolonged reasoning is not all you need: Certainty-based adaptive routing for efficient llm/mllm reasoning. *arXiv preprint arXiv:2505.15154*.

Jinghui Lu, Rui Zhao, Brian Mac Namee, and Fei Tan. 2023a. Punifiedner: A prompting-based unified ner system for diverse datasets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 13327–13335.

Jinghui Lu, Dongsheng Zhu, Weidong Han, Rui Zhao, Brian Mac Namee, and Fei Tan. 2023b. What makes pre-trained language models better zero-shot learners? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2288–2303, Toronto, Canada. Association for Computational Linguistics.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023c. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Daniel Reich and Tanja Schultz. 2024. Uncovering the full potential of visual grounding methods in VQA. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4406–4419, Bangkok, Thailand. Association for Computational Linguistics.

Y. Rubner, C. Tomasi, and L.J. Guibas. 1998. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 59–66.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. *arXiv preprint arXiv:1904.01557*.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Alvin Wan, Xiaoliang Dai, Peizhao Zhang, Zijian He, Yuandong Tian, Saining Xie, Bichen Wu, Matthew Yu, Tao Xu, Kan Chen, et al. 2020. Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12965–12974.

Han Wang, Yongjie Ye, Bingru Li, Yuxiang Nie, Jinghui Lu, Jingqun Tang, Yanjie Wang, and Can Huang. 2025. Vision as lora. *arXiv preprint arXiv:2503.20680*.

Huiming Wang, Liying Cheng, Wenxuan Zhang, De Wen Soh, and Lidong Bing. 2024a. Order-agnostic data augmentation for few-shot named entity recognition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7792–7807, Bangkok, Thailand. Association for Computational Linguistics.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. 2024b. Measuring multimodal mathematical reasoning with math-vision dataset. *arXiv preprint arXiv:2402.14804*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024c. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Qunbo Wang, Ruyi Ji, Tianhao Peng, Wenjun Wu, Zechao Li, and Jing Liu. 2024d. Soft knowledge prompt: Help external knowledge become a better teacher to instruct LLM in knowledge-based VQA. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6132–6143, Bangkok, Thailand. Association for Computational Linguistics.

Haoyang Wen, Eduard Hovy, and Alexander Hauptmann. 2024. Transitive consistency constrained learning for entity-to-entity stance detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1467–1480, Bangkok, Thailand. Association for Computational Linguistics.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. 2012. Detecting texts of arbitrary orientations in natural images. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1083–1090. IEEE.

Qiang Yi, Yangfan He, Jianhui Wang, Xinyuan Song, Shiyao Qian, Xinhang Yuan, Miao Zhang, Li Sun, Keqin Li, Kuan Lu, et al. 2025. Score: Story coherence and retrieval enhancement for ai narratives. *arXiv preprint arXiv:2503.23512*.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.

Haiyang Yu, Jinghui Lu, Yanjie Wang, Yang Li, Han Wang, Can Huang, and Bin Li. 2025. Eve: Towards end-to-end video subtitle extraction with vision-language models. *arXiv preprint arXiv:2503.04058*.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer.

Zhuohao Yu, Chang Gao, Wenjin Yao, Yidong Wang, Wei Ye, Jindong Wang, Xing Xie, Yue Zhang, and Shikun Zhang. 2024. KIEval: A knowledge-grounded interactive evaluation framework for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5967–5985, Bangkok, Thailand. Association for Computational Linguistics.

Liu Yuliang, Jin Lianwen, Zhang Shuaitao, and Zhang Sheng. 2017. Detecting curve text in the wild: New dataset and new solution. *arXiv preprint arXiv:1712.02170*.

Yilun Zhao, Yitao Long, Hongjun Liu, Ryo Kamoi, Linyong Nan, Lyuhao Chen, Yixin Liu, Xiangru Tang, Rui Zhang, and Arman Cohan. 2024.

DocMath-eval: Evaluating math reasoning capabilities of LLMs in understanding long and specialized documents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16103–16120, Bangkok, Thailand. Association for Computational Linguistics.

Shilin Zhou, Zhenghua Li, Yu Hong, Min Zhang, Zhefeng Wang, and Baoxing Huai. 2024. CopyNE: Better contextual ASR by copying named entities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2675–2686, Bangkok, Thailand. Association for Computational Linguistics.

A Implementation

The proposed loss function is incorporated into the model's training objective through linear combination with a weighting coefficient $\lambda = 0.3$. The hyperparameters governing the loss computation are maintained at $\alpha = \beta = \sigma = 0.2$ throughout all experiments, unless otherwise specified. All tasks are trained with a learning rate of 10^{-5} for fine-tuning. Our experiments are conducted based on a widely used open source training repository[1].

B Dataset

This section provides a detailed description of each task along with the corresponding evaluation metrics. The illustrations for each task are presented in Figure 5.

Image Grounding. Grounding task aims to output the bounding box of the corresponding object given a description. We compare on the referring expression comprehension (REC) task on RefCOCO (Lin et al., 2014), RefCOCO+ (Yu et al., 2016) and RefCOCOg (Mao et al., 2016) datasets. The Average Accuracy at $\mathrm{IoU} \geq 0.5$ (Acc@0.5) is used as the evaluation metric.

Scene Text Detection. The scene text detection task focuses on detecting text in natural images. We selected several commonly used datasets: TD500 (Yao et al., 2012), ICDAR2015 (Karatzas et al., 2015), CTW1500 (Yuliang et al., 2017) and Total-Text (Ch'ng and Chan, 2017) for scene text detection tasks. We utilize the identical metric employed in the image grounding task.

Clock Time Recognition. The perception of clock aims to recognize the specific time by images of clocks. We compare the performance of accuracy and time gap on a widely-used TIME (gpiosenka, 2022) dataset. The output are formatted as the label of "2_55", as shown in Figure 6. We use overall accuracy as an metric, and additionally count the time gap between the prediction and the ground truth for further evaluation. For example, the time gap between prediction "4_35" and ground truth "6_20" is 1.75 hours.

Mathematical Reasoning. Completing the mathematical reasoning tasks requires models to understand the context and the image of the mathematical field. We select the MathVista (Lu et al., 2023c) and MathVision (Wang et al., 2024b) datasets to evaluate models. We utilize exact matching accuracy to evaluate math reasoning task.

[1]https://github.com/hiyouga/LLaMA-Factory

Arithmetic Calculations. Calculation task involves training LLMs to perform numerical operations accurately. In this task, the "arithmetic_mix" subset from the widely-used mathematics dataset (Saxton et al., 2019) is used for training and evaluation, which contains 2M training and 10k test items. In this task, exact matching accuracy is applied as the evaluation metric.

C Gumbel Softmax

The Gumbel softmax, also known as Concrete Distribution, is a continuous differentiable approximation to categorical sampling. It replaces the non-differentiable argmax operation with a softmax function and Gumbel noise. Given logits π_i, the Gumbel softmax sample y_i is computed as:

$$y_i = \mathrm{softmax}\left((log(\pi_i) + g_i)/\tau\right),$$

where g_i is the Gumbel noise, which is i.i.d. samples drawn from the $\mathrm{Gumbel}(0, 1)$ distribution, and τ is the temperature parameter.

The Gumbel noise term g_i introduces stochasticity into the sampling process, enabling exploration of the probability space while maintaining differentiability. Moreover, using Gumbel noise also works like regularization, which helps provide gradient information near the decision boundary, to improve generalization ability. The temperature parameter τ controls the sharpness of the distribution: as τ approaches 0, the samples become more discrete and closer to one-hot vectors, while higher temperatures make the distribution more uniform. In our implementation, we use $\tau = 0.1$ to ensure that the results are consistent with the original argmax results.

Gumbel softmax is differentiable as it replaces the discrete argmax with a continuous softmax function, allowing gradients to flow through the sampling process during backpropagation. Thus, Gumbel softmax is widely used in scenarios requiring discrete latent variables in neural networks, such as in VAEs(Jang et al., 2016) or reinforcement learning(Huijben et al., 2022; Wan et al., 2020).

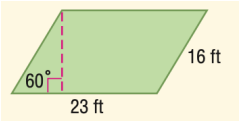
D Preliminaries

This section first briefly introduces the autoregressive decoding process based on cross-entropy in Section D.1, and then compares and analyzes Earth Mover's Distance (EMD) in Section D.2.

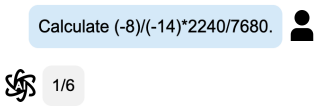
(a) Image Grounding
Question: Where is person bottom left?
Answer: [0.005, 0.332, 0.249, 0.984]

(b) Scene Text Detection
Question: Locate texts in the image.
Answer: [0.388, 0.371, 0.626, 0.440]
[0.418, 0.595, 0.610, 0.653]

(c) Clock Time Recognition
Question: What's the time of this clock?
Answer: 9_20



(d) Mathematic Reasoning
Question: Find the perimeter of the parallelogram.
Answer: 78



(e) Arithmetic Calculation
Question: Calculate (-8)/(-14)*2240/7680.
Answer: 1/6

Figure 5: The illustrations for each task.

D.1 Autoregressive Prediction with Cross Entropy

Autoregressive models operate through sequential decoding, generating tokens one at a time conditioned on previously generated tokens. For each position, the model outputs a probability distribution across the vocabulary, employing the Softmax function to select the most probable token during training.

In the context of language modeling tasks, cross-entropy loss serves as the fundamental training objective for autoregressive models. This loss function quantifies the divergence between the predicted probability distribution and the ground truth distribution:

$$\mathcal{L} = -\sum_{i} p_i \log\left(q_i\right), \tag{5}$$

Åwhere p_i represents the one-hot encoded ground truth distribution, and q_i denotes the model's predicted probability.

While cross-entropy loss effectively minimizes distributional differences between predictions and labels during training, it exhibits a fundamental limitation in autoregressive decoding: the function treats each class independently, disregarding the inherent relationships between different classes. This limitation becomes particularly problematic when modeling numerical sequences where ordinal relationships between values carry semantic significance(Hou et al., 2016), as shown in Figure 2.

D.2 Earth Mover's Distance

To introduce a distance term when calculating the above-mentioned distribution differences, one method is Earth Mover's Distance (EMD), also known as Wasserstein distance. It is an evaluation based on optimal transport theory, measuring the minimal cost of transforming one distribution into the other:

$$\mathrm{EMD}(P, Q) = \min_{\gamma \in \Gamma(P,Q)} \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_{ij} \cdot d\left(x_i, y_j\right), \tag{6}$$

where $P = \{(p_i, x_i)\}$ and $Q = \{(q_i, y_i)\}$ are two discrete distributions, with p_i and q_j are the masses at the points x_i and y_j, respectively. The transport plans, represented as $\Gamma(P, Q)$, are all possible ways to move the mass, and γ_{ij} represents the amount of mass that is transported from p_i to q_j. The distance matrix $d\left(x_i, y_j\right)$ indicates the cost of transporting masses between points x_i and y_j. A widely-used distance matrix d is Euclidean distance.

Since the distance between labels is explicitly considered, predicted values closer to the label are

associated with smaller distance terms. Thus, the Earth Mover's Distance effectively incorporates distance-based weighting. As illustrated in Figure 2, when the distribution is more concentrated around the label, the EMD loss becomes smaller, thereby reflecting the differences between distributions.

D.3 Predicting Digits with EMD

This section presents our approach to refining distance metrics for numerical representation at the digit level. In traditional autoregressive models, cross-entropy loss is typically employed to predict the probability distributions of individual tokens. However, this method treats each numerical digit as an independent entity, disregarding the continuous relationships between numbers. For example, when the target digit is 4, a model prediction of 3 should ideally be considered closer to accurate than a prediction of 9, as it represents a smaller numerical deviation. To address this limitation, we propose incorporating a distance metric that captures these intrinsic numerical relationships more accurately.

Computational Complexity. As established in D.2, Earth Mover's Distance (EMD) provides a robust measure for distributional distances, making it particularly well-suited for numerical prediction tasks. Prior research has applied EMD to align hidden representations within neural networks, often requiring the transport plan (γ_{ij} in Equ. (6)) to be approximated or recalculated dynamically during training. However, the computational demands of EMD present practical challenges, especially in large-scale deep learning applications. Solving the underlying optimization problem in Equ. (6) has a computational complexity of $O\left((n \times m)^3\right)$, which can be prohibitive. Regularized EMD (Cuturi, 2013) addresses this by employing the Sinkhorn-Knopp algorithm to iteratively refine the transport plan γ_{ij} in Equ. (6), reducing complexity to $O\left(k \times n \times m\right)$, where each iteration involves an $O(n \times m)$ matrix operation.

Numerical Prediction Optimization with EMD. When estimating the transport plan, the algorithm's complexity is generally quadratic. However, when restricted to one-dimensional numerical distributions, where the prediction and target values are aligned in position ($i = j$), the transport plan can be simplified to an identity matrix. Thus, Earth Mover's Distance emerges as a highly suitable metric for capturing digit-level numerical distance, formulated as:

$$\mathrm{EMD}(P, Q) = \sum_i |x_i - y_i| \cdot |i - \operatorname{argmax}(Q)|, \tag{7}$$

where the distance matrix $d(x_i, y_j) = |i - \operatorname{argmax}(Q)|$ refers to the index distance of each digit to the label. Given that the predicted probability distribution P is obtained through the softmax transformation, and the ground truth label Q is represented as a one-hot vector, the gradient of EMD with respect to component x_i can be expressed as:

$$\frac{\partial \mathrm{EMD}}{\partial x_i} = \{|k-1|, |k-2|, ..., |k-n|\}. \tag{8}$$

where $k = \operatorname{argmax}(Q)$ denotes the index of the label element in the one-hot vector. This gradient exhibits an inverse relationship with the proximity between the predicted distribution and the ground truth: as the prediction approaches the true label, the magnitude of the gradient diminishes. This characteristic is particularly advantageous for numerical prediction tasks, as it inherently accounts for the ordinal relationships between numerical classes, and addresses the fundamental limitation of the conventional cross-entropy.

$\beta \setminus \alpha$	0	0.1	0.2	0.3	0.4
0	0.780	0.783	0.789	0.789	0.771
0.1	0.777	**0.790**	**0.799**	0.789	0.778
0.2	0.782	**0.793**	**0.795**	0.785	0.784
0.3	0.788	0.783	0.786	0.785	0.781
0.4	0.774	0.782	0.778	0.781	0.784

Table 7: Ablation studies of hyper-parameters α and β

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
σ	0.784	0.785	**0.789**	**0.788**	0.783	0.780	0.770	0.763
λ	0.741	0.777	**0.798**	**0.795**	0.788	0.786	0.787	0.777

Table 8: Ablation studies of the hyper-parameters σ (exp-weighting) and λ (coefficient)

E Ablations

We have supplemented our work with comprehensive experiments on the clock time recognition task using Qwen2-VL-7B, as shown in Table 7 and Table 8. We have bolded some of the outperforming results. The results demonstrate that NTIL adapts well to different hyperparameters.

We have also supplemented NTIL with the GSM8k math reasoning dataset. NTIL brings consistent performance improvement on mixed text-numeric tasks, which shows the effectiveness of our method.

Model	Params	CE	EMD	**NTIL**
Qwen2.5	2b	0.509	0.517	**0.523**
MiniCPM3	4b	0.672	0.676	**0.704**
Yi	6b	0.375	0.383	**0.400**
LLaMA3	8b	0.638	0.643	**0.646**

Table 9: Results of the GSM8k mathematical reasoning task.

F Qualitative Examples

Visualizations of the outputs of different losses are shown in Figure 6, and the examples are taken from experimental results using LLaVA-1.5. For image grounding task (Figure 6(a)), the task was to predict the location of "horse back left" in an image. The CE loss (blue box) performed poorly, with predictions far from the ground truth. EMD (red box) showed an improvement, capturing spatial features better, while NTIL (green box) provided the most accurate predictions, closely matching the ground truth (black box). Overall, NTIL outperformed both CE and EMD, demonstrating its effectiveness in this task.

Figure 6(b) presents a qualitative comparision for clock time recognition task. In this case, NTIL provides the most accurate prediction of the clock time, correctly identifying 2:55, which matches the ground truth. EMD performs better than CE, predicting 2:50, but it is still slightly off. CE, however, predicts 5:10, a significant deviation. Overall, NTIL outperforms both EMD and CE in predicting the clock time accurately.

Question: Where is the horse back left?

CE Prediction: [0.0, 0.138, 0.174, 0.754]

EMD Prediction: [0.447, 0.348, 0.687, 0.875]

NTIL Prediction: [0.567, 0.342, 0.77, 0.774]

Ground Truth: [0.581, 0.34, 0.757, 0.816]

(a) Example in Image Grounding. Blue box is CE prediction, red box is EMD prediction, green box is NTIL prediction. Black box is ground truth.

Question: What's the time of this clock?

CE Prediction: 5_10

EMD Prediction: 2_50

NTIL Prediction: 2_55

Ground Truth: 2_55

(b) Example in clock time recognition.

Figure 6: Comparisons between CE, EMD and NTIL.

FEAT: A Preference Feedback Dataset through a Cost-Effective Auto-Generation and Labeling Framework for English AI Tutoring

Hyein Seo[1], **Taewook Hwang**[1], **Yohan Lee**[2], **Sangkeun Jung**[13*]
[1]Chungnam National University
[2]Electronics and Telecommunications Research Institute
[3]EurekaAI
{hyenee97,taewook5295}@gmail.com, carep@etri.re.kr, hugmanskj@gmail.com

Abstract

In English education tutoring, teacher feedback is essential for guiding students. Recently, AI-based tutoring systems have emerged to assist teachers; however, these systems require high-quality and large-scale teacher feedback data, which is both time-consuming and costly to generate manually. In this study, we propose FEAT, a cost-effective framework for generating teacher feedback, and have constructed three complementary datasets[1]: (1) DIRECT-Manual (DM), where both humans and large language models (LLMs) collaboratively generate high-quality teacher feedback, albeit at a higher cost; (2) DIRECT-Generated (DG), an LLM-only generated, cost-effective dataset with lower quality;, and (3) DIRECT-Augmented (DA), primarily based on DG with a small portion of DM added to enhance quality while maintaining cost-efficiency. Experimental results showed that incorporating a small portion of DM (5–10%) into DG leads to superior performance compared to using 100% DM alone.

1 Introduction

In English education tutoring, providing appropriate teacher feedback plays a crucial role in guiding students and improving their educational outcomes (Ma et al., 2014; Fossati, 2008). Given its importance, various studies have explored automated teacher feedback generation (Meyer et al., 2024; Scarlatos et al., 2024b; Liu et al., 2020).

Figure 1 illustrates methods for generating and annotating teacher feedback for tutoring systems. As shown in (a), human-generated feedback with ranking provides high quality, but its time-consuming and costly nature makes it difficult to scale up (Chang et al., 2023).

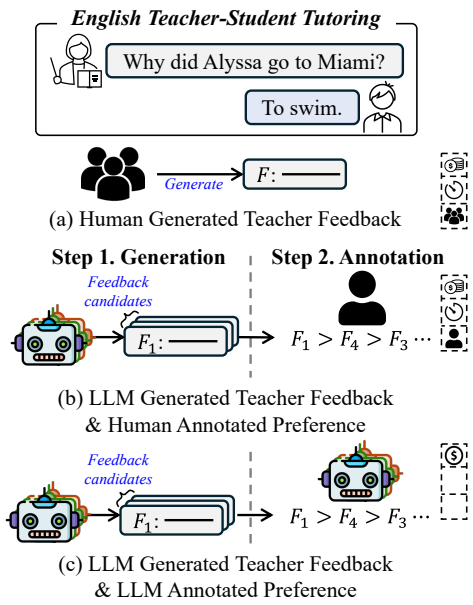


Figure 1: Teacher feedback generation and annotation process in an English tutoring system.

To address this challenge, we propose **FEAT**, a cost-effective framework using large language models (LLMs) to automatically generate a large-scale teacher feedback preference dataset for tutoring AI. This enables reward- or rank-based learning, making it suitable for building human-friendly tutoring models (Ouyang et al., 2022). FEAT generates teacher feedback based on student responses, using the dialogue history between teacher and student and context as input. Moreover, we apply feedback criteria defined by Seo et al. (2025) to ensure educationally appropriate feedback.

Using FEAT, we constructed three datasets: (1) *DIRECT-Manual* (DM), which contains human and LLM-generated feedback with human-annotated rankings (high quality and high cost), (2) *DIRECT-Generated* (DG), an entirely LLM-generated and annotated preference dataset (medium quality and low cost), and (3) *DIRECT-Augmented* (DA), a hybrid dataset built on DG with a minor addition of DM (high quality and low cost).

*Corresponding author
[1]Our dataset is publicly available at https://github.com/hyenee/FEAT

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 575–589
July 27 - August 1, 2025

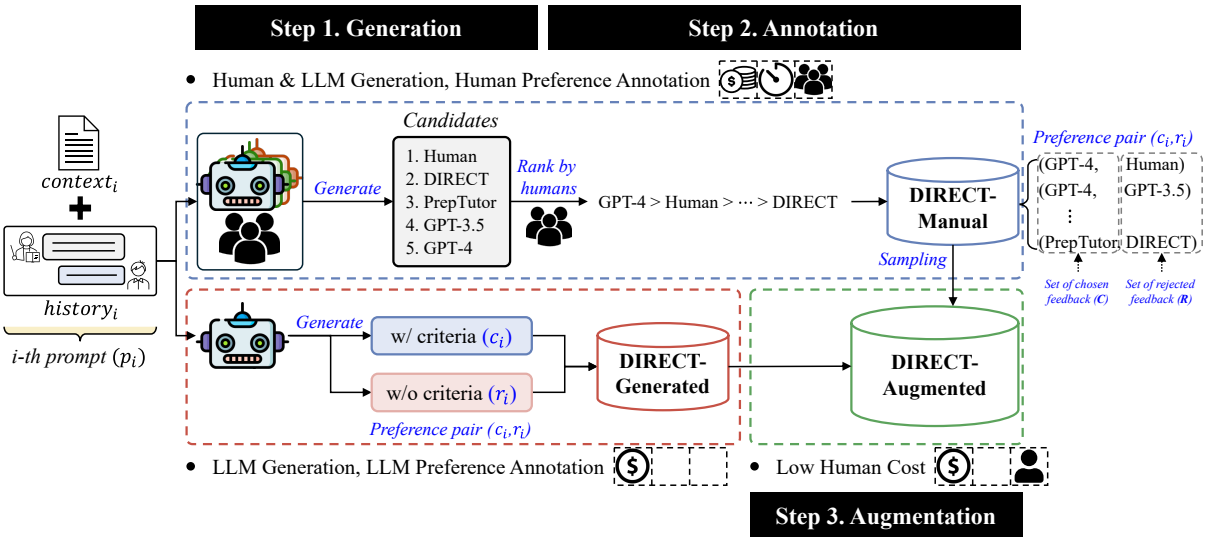


Figure 2: The architecture of the FEAT framework, illustrating the construction process of the DIRECT-Manual, DIRECT-Generated, and DIRECT-Augmented datasets. p_i, c_i, and r_i denote the i-th prompt, chosen, and rejected responses, respectively.

Our experiments showed that incorporating a small portion of DM (5–10%) into DG leads to superior performance compared to using DM alone.

Our main contributions are as follows:

- We proposed FEAT, a cost-effective framework for automated teacher feedback generation and annotation in English tutoring.
- We constructed three preference datasets: DM, DG, and DA, enabling reward- or rank-based learning.
- We confirmed that incorporating a small amount of DM into DG (DA) yielded better performance than DM alone.

2 FEAT: *F*eedback Dataset Generation Framework for *E*nglish *A*I *T*utoring

Figure 2 illustrates the construction process of our FEAT framework. FEAT applies five criteria from Seo et al. (2025)-*Correct, Revealing, Guidance, Diagnostic,* and *Encouragement*—ensuring educationally effective feedback.

2.1 DIRECT-Manual: Rank-based Preference Dataset

DM is an extended version of the DIRECT (Huang et al., 2023) dataset, simulating intelligent tutoring between teachers and students. While it ensures high quality, it relies heavily on human effort, making it time-consuming and costly.

Step 1: Feedback Generation. We collected teacher feedback data for scenarios with incorrect student answers in teacher-student dialogues from diverse sources (Human, DIRECT, PrepTutor, GPT-3.5, and GPT-4; see Appendix B), with DM previously used as private data in Liermann et al. (2024). An example from the DM is shown in Figure 3.

Step 2: Feedback Ranking via Human Annotation. Human annotators ranked the five feedback candidates using two criteria: *Correct* (specific and accurate information) and *Revealing* (avoiding direct answers). Feedback meeting both criteria received the highest rank, with Correct prioritized when only one criterion was met.

Step 3: Preference Data Construction. From ranked five feedback candidates, we created pairwise combinations; in each pair, the higher-ranked feedback is labeled ***chosen*** and the lower-ranked as ***rejected***.

2.2 DIRECT-Generated: Criteria-based Preference Dataset

DG uses LLM to automatically generate and annotate teacher feedback based on specific criteria, producing reasonably good data at a lower cost.

Step 1: Feedback Generation. Using dialogue history and context, LLM generates teacher feedback based on five criteria. We created tutoring scenarios by converting reading comprehension tasks from MCTest (Richardson et al., 2013) to generate large-scale feedback data. A sample from the MCTest is illustrated in Figure 4.

Step 2: Preference Data Construction. We generated two types of feedback: *w/ criteria* (applying

Story
Bill is ten years old.
One day his friend Tom says to him, "I am going to have a birthday party on Sunday evening. Bill, can you come to my party? " "I'll be glad to," answers Bill.
...

History
Teacher : All right. Do you know how old Bill is?
Student : He is eight years old.

Reference
Ten. (Bill is ten years old.)

Feedback Candidates
- Human: Pretty close, but he is a bit older than that.
- DIRECT: no, he is one year older than that.
- PrepTutor: can you think more? honestly, bill is now 10 years old and he still has lots of growth. how old is he?
- GPT-3.5: Hmm, that's close. But is there any information in the passage that tells us Bill's age?
- GPT-4: Take a closer look at the beginning of the passage. It mentions Bill's age there.

Ranking
GPT-4>Human>GPT-3.5>PrepTutor>Tutor

Figure 3: Sample from the DIRECT-Manual.

	Train	Test
DIRECT-Manual	5,025	475
DIRECT-Generated	3,996	444

Table 1: Dataset statistics.

five criteria) and *w/o criteria* (without criteria). We labeled *w/ criteria* feedback as ***chosen*** and *w/o criteria* as ***rejected***, assuming criteria-based feedback is of higher quality. Data statistics are shown in Table 1, with full details in Appendix C.

3 Teacher Feedback Ranking

To validate the preference annotations in DM, DG, and DA, we trained pairwise-based ranking models.

3.1 Ranking Models

We employed five approaches to train ranking models. Each model takes *(prompt, chosen, rejected)* as input.

Binary Classifier formulates preference learning as a binary classification task, labeling *(chosen, rejected)* pairs as 1 and *(rejected, chosen)* pairs as 0. The input sequence is depicted in Figure 11.

Reward Model (Ouyang et al., 2022) computes scalar preference scores for feedback pairs, training to assign higher scores to chosen feedback.

Direct Preference Optimization (DPO) (Rafailov et al., 2023) optimizes language model probabilities to prefer chosen feedback, using log probability differences between chosen and rejected pairs.

RankNet (Burges et al., 2005) learns score differences between feedback pairs using Binary Cross-Entropy loss. Reward Model, DPO, and RankNet share the same prompt, shown in Figure 12.

Ensemble aggregates predictions from the above four approaches through majority voting.

Story
Alyssa got to the beach after a long trip. She's from Charlotte. She traveled from Atlanta. She's now in Miami. She went to Miami to visit some friends. But she wanted some time to herself at the beach, so she went there first. After going swimming and laying out, she went to her friend Ellen's house.
...

Question
Why did Alyssa go to Miami?

Answer Options
A) Swim
B) Travel
C) Visit friends
D) Laing out

Student Correct Response
The answer is visit friends.

Student Incorrect Response
The answer is swim.

Figure 4: Sample from the MCTest.

3.2 Scenarios for Training

We evaluated our ranking models on DM with three training configurations. The arrow (→) indicates training (left) and evaluation (right).

- **DM→DM**: Training with DM using manual annotation, serving as a performance upper bound for comparison with the other two scenarios (DG→DM and DA→DM).
- **DG→DM**: Training with DG using automatic annotation.
- **DA→DM**: Hybrid training using DG combined with a subset of DM for mixed annotation.

During training, we enhanced data diversity by including feedback from different contexts beyond the standard *(chosen, rejected)* pairs. This approach enabled the model to learn feedback comparisons across various contexts.

For evaluation, we tested the model on all possible pairs in DM. The model's pairwise predictions

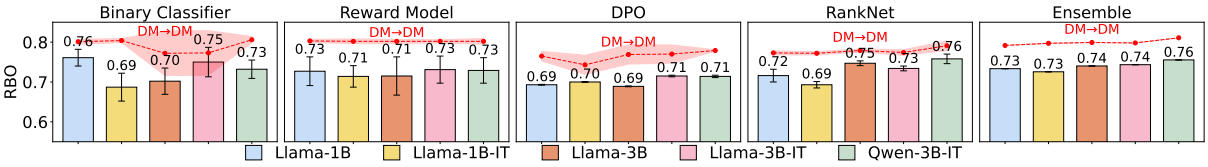
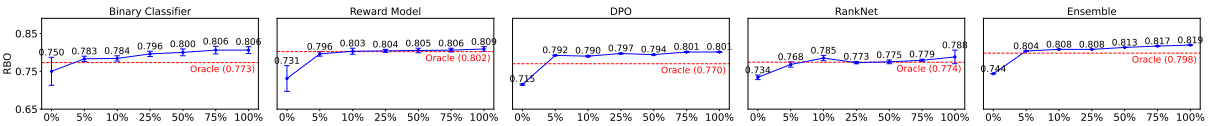
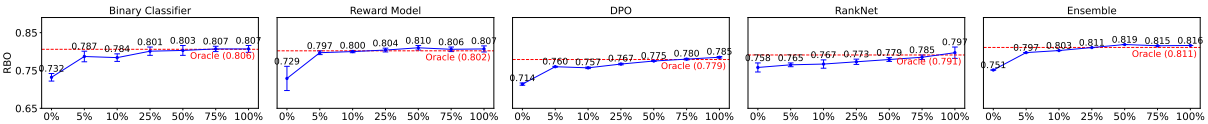


Figure 5: Ranking model performance across different approaches (with 5-seed standard deviation). Lines indicate DM→DM performance, while bars show DG→DM performance.



(a) Llama-3B-IT



(b) Qwen-3B-IT

Figure 6: Llama-3B-IT and Qwen-3B-IT performance in the DA→DM scenario. (See Appendix E for other models.)

were aggregated to create overall rankings, with accuracy scored as 1 for *chosen* > *rejected* and 0 for *chosen* < *rejected*.

4 Experiments

We designed experiments and analyzed results to address the following research questions:

- How does ranking model performance with DG compare to human-curated DM (Section 4.2)?
- How does the ratio of DM in DA affect ranking model performance (Section 4.3)?
- How does the number of criteria in DG affect ranking models performance? (Section 4.4)?

4.1 Experimental Setup

Models We trained ranking models using five open-source models: Llama-1B (Dubey et al., 2024), Llama-1B-IT, Llama-3B, Llama-3B-IT, and Qwen-3B-IT (Bai et al., 2023). Model details and hyper-parameters are provided in Appendix D.

Evaluation Metrics Rank-biased overlap (RBO) is a metric used to measure the similarity between two ranked lists. It ranges from 0 to 1, with values closer to 1 indicating higher similarity between the lists.

4.2 Comparison of Ranking Model Performance

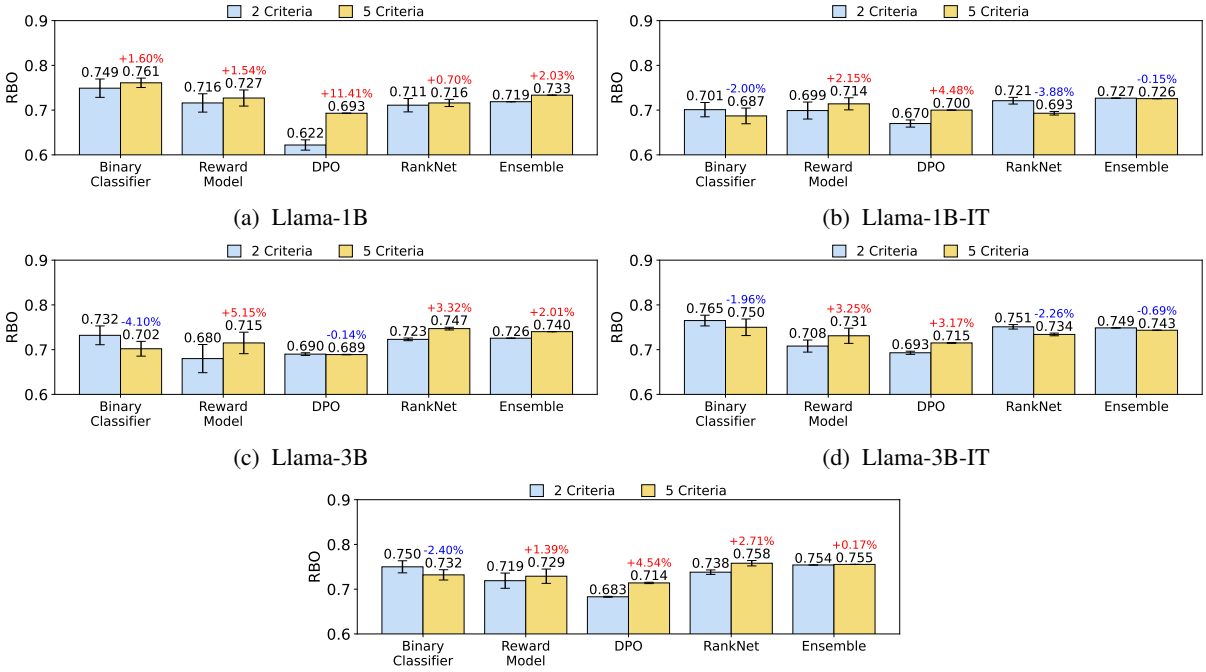
As shown in Figure 5, DM→DM performed consistently (0.77-0.80) across all sizes and approaches. While DG→DM showed lower but competitive results: the binary classifier reached 0.76 (Llama-1B), reward model 0.73 (Llama-3B-IT), DPO 0.71 (Llama-3B-IT, Qwen-3B-IT), RankNet 0.76 (Qwen-3B-IT), and Ensemble 0.76 (Qwen-3B-IT). Notably, the Ensemble maintained stable performance across different architectures, mitigating the variability seen in individual approaches.

These results indicated that teacher feedback generated by LLMs can produce rankings highly comparable to human annotator rankings, with a particularly strong trend observed in larger models.

Case Study As a result of analyzing the RBO scores between the ground-truth and predicted rankings, Figure 14 demonstrates that the two rankings are nearly identical, except for the swapped positions of DIRECT and PrepTutor, resulting in an RBO score of 0.8833. In contrast, Figure 15 exhibits significantly lower agreement between the ground-truth and predicted rankings, with an RBO score of 0.4166. In DM, which is limited to five feedback candidates, the RBO score maintains a baseline similarity of at least 0.4 even when the rankings are completely different, due to the limited number of possible permutations. Additional experimental results are provided in Appendix E.

4.3 Performance Analysis by DM Ratio in the DA→DM Scenario

We analyzed how varying the proportion of DM (5-100%) in the DA→DM scenario affects model per-



(a) Llama-1B

(b) Llama-1B-IT

(c) Llama-3B

(d) Llama-3B-IT

(e) Qwen-3B-IT

Figure 7: Overall performance across varying numbers of feedback criteria.

formance. Figure 6 presents the results for Llama-3B-IT and Qwen-3B-IT, the models that achieved the strongest performance under most approaches.

For Llama-3B-IT, the binary classifier, DPO, and Ensemble outperformed the DM→DM even with only 5% of human-annotated DM and DA data. Similarly, Reward Model and RankNet exceeded DM→DM performance within the 5–10% annotation range. In contrast, Qwen-3B-IT surpassed DM→DM primarily within the 50–75% or 75–100% annotation ranges. Although Qwen-3B-IT is not as efficient as Llama-3B-IT, the results suggest that high performance can be achieved with minimal human annotation costs. The overall performance of Llama-1B, Llama-1B-IT, and Llama-3B models is illustrated in Figure 13 (see Appendix E).

4.4 Performance Analysis by Number of Feedback Criteria

To investigate the impact of feedback criteria, we compared two training configurations: using all five criteria versus using only two essential criteria (Correct and Revealing). For the experiments, we generated an additional version of DG that includes only two feedback criteria and trained a ranking model in the DG→DM scenario.

Figure 7 shows that increasing the number of feedback criteria from two to five consistently improves Llama-1B across all approaches, with DPO exhibiting the largest gain (+11.41 %). For Qwen-3B-IT, every approach except the binary classifier benefits from the richer feedback, and the remaining models display improvements in selected approaches. These results suggest that incorporating richer feedback information enhances model generalization.

5 Conclusion

In this study, we proposed the **F**eedback Dataset Generation Framework for **E**nglish **AI** **T**utoring (FEAT), which utilizes LLMs to generate teacher feedback and build preference datasets for English tutoring. We evaluated ranking models on three datasets—DIRECT-Manual (DM), DIRECT-Generated (DG), and DIRECT-Augmented (DA)—constructed via FEAT.

Results showed that models based on DG performed competitively with DM-based models. Moreover, supplementing with only 5–10% human-annotated DM led to superior performance than using the full DM dataset. These findings demonstrate that high performance can be achieved with minimal human effort with our FEAT framework. In future research, we will extend our framework to broader educational scenarios.

Limitations

In this study, we explored the feasibility of LLM-based teacher feedback generation and preference dataset construction using the FEAT framework. However, the study has the following limitations:

First, while we constructed an English tutoring scenario using the MCTest dataset, further research is required to assess the generalizability of the framework across diverse educational datasets.

Second, we only conducted the ranking model experiments using 1B and 3B LLMs. Future work should explore the applicability of larger LLMs (e.g., 7B, 13B, 70B) to evaluate their impact on ranking performance.

Third, we employed a pairwise approach for ranking model training. We plan to explore preference dataset construction and training strategies applicable to pointwise and listwise ranking approaches in future research.

Acknowledgement

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2019-0-00004, Development of semisupervised learning language intelligence technology and Korean tutoring service for foreigners), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-00556217314820926401 01), and research fund of Chungnam National University.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Nischal Ashok Kumar and Andrew Lan. 2024. Improving socratic question generation using data augmentation and preference optimization. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 108–118, Mexico City, Mexico. Association for Computational Linguistics.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 89–96, New York, NY, USA. Association for Computing Machinery.

Christopher Burges, Robert Ragno, and Quoc Le. 2006. Learning to rank with nonsmooth cost functions. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 129–136, New York, NY, USA. Association for Computing Machinery.

Nadine Chang, Francesco Ferroni, Michael J Tarr, Martial Hebert, and Deva Ramanan. 2023. Thinking like an annotator: Generation of dataset labeling instructions. *arXiv preprint arXiv:2306.14035*.

Koby Crammer and Yoram Singer. 2001. Pranking with ranking. In *Advances in Neural Information Processing Systems*, volume 14. MIT Press.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. 2023. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*.

Wei Dai, Jionghao Lin, Hua Jin, Tongguang Li, Yi-Shan Tsai, Dragan Gašević, and Guanliang Chen. 2023. Can large language models provide feedback to students? a case study on chatgpt. In *2023 IEEE International Conference on Advanced Learning Technologies (ICALT)*, pages 323–325.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Wanyong Feng, Jaewook Lee, Hunter McNichols, Alexander Scarlatos, Digory Smith, Simon Woodhead, Nancy Ornelas, and Andrew Lan. 2024. Exploring automated distractor generation for math multiple-choice questions via large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3067–3082, Mexico City, Mexico. Association for Computational Linguistics.

Davide Fossati. 2008. The role of positive feedback in intelligent tutoring systems. In *Proceedings of the ACL-08: HLT Student Research Workshop*, pages 31–36.

Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov):933–969.

Wensheng Gan, Zhenlian Qi, Jiayang Wu, and Jerry Chun-Wei Lin. 2023. Large language models in education: Vision and opportunities. In *2023 IEEE International Conference on Big Data (BigData)*, pages 4776–4785.

John Hattie and Helen Timperley. 2007. The power of feedback. *Review of educational research*, 77(1):81–112.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jin-Xia Huang, Yohan Lee, and Oh-Woog Kwon. 2023. Direct: Toward dialogue-based reading comprehension tutoring. *IEEE Access*, 11:8978–8987.

Jaeho Jeon and Seongyong Lee. 2023. Large language models in education: A focus on the complementary relationship between human teachers and chatgpt. *Education and Information Technologies*, 28(12):15873–15892.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 133–142, New York, NY, USA. Association for Computing Machinery.

Gerd Kortemeyer. 2024. Performance of the pre-trained large language model gpt-4 on automated short answer grading. *Discover Artificial Intelligence*, 4(1):47.

Charles Koutcheme, Nicola Dainese, Arto Hellas, Sami Sarsa, Juho Leinonen, Syed Ashraf, and Paul Denny. 2024. Evaluating language models for generating and judging programming feedback.

Yanyan Lan, Yadong Zhu, Jiafeng Guo, Shuzi Niu, and Xueqi Cheng. 2014. Position-aware listmle: a sequential learning process for ranking. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, page 449–458, Arlington, Virginia, USA. AUAI Press.

Yu-Ju Lan and Nian-Shing Chen. 2024. Teachers' agency in the era of llm and generative ai. *Educational Technology & Society*, 27(1):I–XVIII.

Jaewook Lee, Digory Smith, Simon Woodhead, and Andrew Lan. 2024. Math multiple choice question generation via human-large language model collaboration. In *Proceedings of the 17th International Conference on Educational Data Mining*, pages 941–946, Atlanta, Georgia, USA. International Educational Data Mining Society.

Ping Li, Qiang Wu, and Christopher Burges. 2007. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

Wencke Liermann, Jin-Xia Huang, Yohan Lee, and Kong Joo Lee. 2024. More insightful feedback for tutoring: Enhancing generation mechanisms and automatic evaluation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10838–10851, Miami, Florida, USA. Association for Computational Linguistics.

Haochen Liu, Zitao Liu, Zhongqin Wu, and Jiliang Tang. 2020. Personalized multimodal feedback generation in education. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1826–1840, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Haohao Luo, Yang Deng, Ying Shen, See-Kiong Ng, and Tat-Seng Chua. 2024a. Chain-of-exemplar: Enhancing distractor generation for multimodal educational question generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7978–7993, Bangkok, Thailand. Association for Computational Linguistics.

Jian Luo, Xuanang Chen, Ben He, and Le Sun. 2024b. PRP-graph: Pairwise ranking prompting to LLMs with graph aggregation for effective text re-ranking. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5766–5776, Bangkok, Thailand. Association for Computational Linguistics.

Wenting Ma, Olusola O Adesope, John C Nesbit, and Qing Liu. 2014. Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of educational psychology*, 106(4):901.

Jennifer Meyer, Thorben Jansen, Ronja Schiller, Lucas W. Liebenow, Marlene Steinbach, Andrea Horbach, and Johanna Fleckenstein. 2024. Using llms to bring evidence-based feedback into the classroom: Ai-generated feedback increases secondary students' text revision, motivation, and positive emotions. *Computers and Education: Artificial Intelligence*, 6:100199.

Nikahat Mulla and Prachi Gharpure. 2023. Automatic question generation: a review of methodologies, datasets, evaluation metrics, and applications. *Progress in Artificial Intelligence*, 12(1):1–32.

David J Nicol and Debra Macfarlane-Dick. 2006. Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in higher education*, 31(2):199–218.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John

Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518, Mexico City, Mexico. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA. Association for Computational Linguistics.

Alexander Scarlatos, Wanyong Feng, Andrew Lan, Simon Woodhead, and Digory Smith. 2024a. Improving automated distractor generation for math multiple-choice questions with overgenerate-and-rank. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 222–231, Mexico City, Mexico. Association for Computational Linguistics.

Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2024b. Improving the validity of automatically generated feedback via reinforcement learning.

Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2024c. Improving the validity of automatically generated feedback via reinforcement learning. In *International Conference on Artificial Intelligence in Education*, pages 280–294. Springer.

Hyein Seo, Taewook Hwang, Jeesu Jung, Hyeonseok Kang, Hyuk Namgoong, Yohan Lee, and Sangkeun Jung. 2025. Large language models as evaluators in education: Verification of feedback consistency and accuracy. *Applied Sciences*, 15(2):671.

Valerie J Shute. 2008. Focus on formative feedback. *Review of educational research*, 78(1):153–189.

Maja Stahl, Leon Biermann, Andreas Nehring, and Henning Wachsmuth. 2024. Exploring LLM prompting strategies for joint essay scoring and feedback generation. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 283–298, Mexico City, Mexico. Association for Computational Linguistics.

Jacob Steiss, Tamara Tate, Steve Graham, Jazmin Cruz, Michael Hebert, Jiali Wang, Youngsun Moon, Waverly Tseng, Mark Warschauer, and Carol Booth Olson. 2024. Comparing the quality of human and chatgpt feedback of students' writing. *Learning and Instruction*, 91:101894.

Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka. 2008. Softrank: optimizing non-smooth rank metrics. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, page 77–86, New York, NY, USA. Association for Computing Machinery.

Ming-Feng Tsai, Tie-Yan Liu, Tao Qin, Hsin-Hsi Chen, and Wei-Ying Ma. 2007. Frank: a ranking method with fidelity loss. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 383–390, New York, NY, USA. Association for Computing Machinery.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S Yu, and Qingsong Wen. 2024. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105*.

Zichao Wang, Jakob Valdez, Debshila Basu Mallick, and Richard G. Baraniuk. 2022. Towards human-like educational question generation with large language models. In *Artificial Intelligence in Education*, pages 153–166, Cham. Springer International Publishing.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Zhaohui Zheng, Keke Chen, Gordon Sun, and Hongyuan Zha. 2007. A regression framework for learning ranking functions using relative relevance judgments. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 287–294, New York, NY, USA. Association for Computing Machinery.

A Related Works

A.1 Feedback in Education

In the field of education, teacher feedback plays a crucial role in enhancing students' learning experiences and achievements. In particular, immediate and appropriate feedback positively impacts students' cognitive, emotional, and motivational outcomes (Shute, 2008).

Research has been conducted on designing effective feedback strategies. Nicol and Macfarlane-Dick (2006) proposed seven principles for effective feedback, while Hattie and Timperley (2007) analyzed the impact of feedback on learning and investigated its key components. Steiss et al. (2024); Scarlatos et al. (2024c) proposed five criteria for evaluating feedback quality, designing them to help students understand clear directions for improvement and maintain motivation. Additionally, research applying feedback criteria has been conducted in fields such as programming education (Koutcheme et al., 2024).

A.2 Large Language Models in Education

Advancements in large language models (LLMs) have significantly impacted the field of education (Gan et al., 2023; Wang et al., 2024; Lan and Chen, 2024; Jeon and Lee, 2023; Dai et al., 2023). The integration of LLMs with educational technology has been applied across various domains, including automated short answer grading (Kortemeyer, 2024), automated essay scoring (Stahl et al., 2024), automated distractor generation (Feng et al., 2024; Scarlatos et al., 2024a), and automatic question generation (Luo et al., 2024a; Lee et al., 2024; Ashok Kumar and Lan, 2024; Mulla and Gharpure, 2023; Wang et al., 2022).

Research has also been conducted on automated feedback systems to provide better feedback to students (Dai et al., 2023; Meyer et al., 2024). Additionally, LLM-powered personalized feedback generation has contributed to reducing teachers' workloads and improving the efficiency of online education (Liu et al., 2020). Beyond feedback generation, LLMs have also been utilized for feedback quality assessment. Studies have proposed LLM-based feedback generation and evaluation frameworks in domains such as programming assignments (Koutcheme et al., 2024) and mathematics (Scarlatos et al., 2024c), demonstrating the potential of LLMs in educational settings.

A.3 Learning to Rank Approaches

Learning to Rank (LTR) is widely used in information retrieval and recommendation systems, aiming to learn the optimal ranking of items for a given query. LTR methodologies are generally categorized into three approaches: Pointwise, Pairwise, and Listwise.

Pointwise approaches, such as MCRank (Li et al., 2007) and PRank (Crammer and Singer, 2001), predict a relevance score for each item individually and rank them based on these scores. Pairwise approaches learn the relative preference between two items, with representative algorithms including RankNet (Burges et al., 2005), LambdaRank (Burges et al., 2006), RankSVM (Joachims, 2002), RankBoost (Freund et al., 2003), GBRank (Zheng et al., 2007), and FRank (Tsai et al., 2007). Listwise approaches consider the entire item list as a single input and optimize its order holistically, with prominent algorithms such as ListNet (Cao et al., 2007), ListMLE (Lan et al., 2014), and SoftRank (Taylor et al., 2008).

Recent LTR research has expanded to leverage LLMs for ranking tasks (Cui et al., 2023). Qin et al. (2024); Luo et al. (2024b) proposed LLM-based pairwise ranking methods, demonstrating the potential of large-scale language models in ranking optimization.

	Human	DIRECT	PrepTutor	GPT-3.5	GPT-4
Word	10.02	9.27	29.37	17.84	21.01
Token	13.98	13.25	36.60	22.61	26.58

Table 2: Average feedback word length and token length in the DIRECT-Manual dataset.

	# Data	Average Words Per:	
		Story	Question
DIRECT-Manual	5,500	193.05	11.90
MCTest	1,480	202.71	7.79

Table 3: DIRECT-Manual and MCTest dataset statistics.

B Details of DIRECT-Manual Dataset

B.1 Feedback Generation Process

The DIRECT-Manual (DM) consists of five feedback candidates, each generated through different methods. The details of these candidates are as follows:

- **Human**: Feedback written by human annotators.
- **DIRECT**: Feedback generated using GPT-2 trained on the DIRECT dataset.
- **PrepTutor**: Feedback generated using GPT-2 fine-tuned on external domain-specific feedback data.
- **GPT-3.5**: Feedback generated using GPT-3.5-turbo-0613.

Prompt for generating DIRECT-Manual

You are a proficient tutoring assistant who provides just a few clues to an user in the correct direction.
The user should understand the following passage and then answer your question.

Passage: **{passage}**

The correct answer is "{correct answer}", but the user don´t answer correctly as the following tutoring dialogues.
Generate an indirect feedback or hint to guide the user to find the answer on him/her own.

{student & teacher dialogue}

Figure 8: Prompt for generating DIRECT-Manual.

- **GPT-4**: Feedback generated using GPT-4-0613.

B.2 Feedback Ranking Process

The feedback candidates were ranked by human annotators based on two criteria:

- **Correct**: The feedback provides specific factual information based on the student's response or the given text.
- **Revealing**: The feedback guides the student toward the correct answer without explicitly stating it.

Table 2 presents the average length of feedback candidates, while Table 3 provides overall dataset statistics. Figure 8 illustrates the prompt used for feedback generation with GPT-3.5 and GPT-4 in the DM.

C Details of DIRECT-Generated Dataset

C.1 Dataset Preprocessing Process

The DIRECT-Generated (DG) dataset was constructed based on MCTest (Richardson et al., 2013), which consists of stories designed for students in grades 1–4, along with corresponding questions and four answer options.

The dataset construction involved the following preprocessing steps:

1. The question field from MCTest was used as the teacher's question.
2. The answer field was used as the student's correct response.
3. One option from the answer choices was randomly selected as the student's incorrect response.

C.2 Feedback Generation Process

We utilized LLMs to automatically generate teacher feedback. The prompt for feedback generation was designed to include the story, question, the student's incorrect response, and the correct response. Additionally, the five feedback criteria defined by Seo et al. (2025) were applied to ensure educationally effective feedback. The characteristics of each criterion are as follows:

- **Correct**: The feedback should be factually accurate and directly related to the student's response and the question.
- **Revealing**: The feedback should avoid explicitly providing the correct answer to the student.
- **Guidance**: The feedback should offer direction or hints to help the student progress towards the right answer.
- **Diagnostic**: The feedback should pinpoint and address any misconceptions or errors made by the student.
- **Encouragement**: The feedback should convey a positive and supportive tone to motivate the student.

The following LLMs were used for feedback generation:

- GPT-4o (Achiam et al., 2023)
- Claude-3[2] (Bai et al., 2022)
- Llama-3.1-70B-Instruct[3] (Dubey et al., 2024)

Figures 9 and 10 illustrate example prompts. The prompts were designed to generate teacher feedback that guides students from incorrect to correct responses. The feedback generated by all three LLMs was aggregated and then split into train and test datasets at a 9:1 ratio.

Table 4 presents examples of teacher feedback generated under different prompt strategies in the

[2] claude-3-5-sonnet-20240620

[3] https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct

DG dataset. Notably, when feedback criteria were not applied (w/o criteria), the generated feedback often explicitly stated the correct answer. In contrast, when feedback criteria were applied (w/ criteria), the generated feedback was more structured and pedagogically aligned.

D Implementation Details

All experiments were conducted in NVIDIA A100 (40GB VRAM) GPUs and implemented using the PyTorch. The Hugging Face (Wolf et al., 2019) was utilized for model training. All models were fine-tuned using the Low-Rank Adaptation (LoRA) (Hu et al., 2021). The versions of the models used are listed in Table 5, and detailed hyperparameter settings are provided in Table 6. The input format of the ranking model is illustrated in Figure 11 and Figure 12.

E Additional Experimental Results

Table 7 presents the overall performance across different ranking model approaches. All experiments were conducted five runs with different random seeds.

Figure 13 summarizes the results of the DIRECT-A (DA) →DIRECT-M (DM) scenario described in Section 4.3 for the Llama-1B, Llama-1B-IT, and Llama-3B. In most ranking model approaches, performance improved as the proportion of DM increased. For every model, DPO exceeds the DM→DM baseline even when trained with only 0–5% of the DM data. Notably, Llama-3B surpasses the DM→DM baseline in all methods with at most 10–25% of the DM data.

Figures 14 and 15 show examples of high and low RBO scores between ground-truth rankings and predicted rankings, respectively. When the two rankings are nearly identical, the RBO reaches 0.8333; when they diverge markedly, the score drops to 0.4166.

Prompt for generating the teacher's feedback without criteria provided

Instruction
You are an English teacher tasked with providing feedback to students.
Your goal is to provide feedback that guides the student from an incorrect answer to the correct one.
The feedback must be limited to one sentence.

Format
Respond in JSON format with the following structure:
{{
  "feedback": "Your feedback here."
}}

Story
{story}

Question
{question}

Incorrect Response
{student incorrect response}

Correct Response
{student correct response}

Figure 9: Prompt for generating the teacher's feedback with criteria provided.

Prompt for generating the teacher's feedback without criteria provided

Instruction
You are an English teacher tasked with providing feedback to students.
Your goal is to provide feedback that guides the student from an incorrect answer to the correct one.
The feedback must be limited to one sentence.

Format
Respond in JSON format with the following structure:
{{
  "feedback": "Your feedback here."
}}

Story
{story}

Question
{question}

Incorrect Response
{student incorrect response}

Correct Response
{student correct response}

Figure 10: Prompt for generating the teacher's feedback without criteria provided.

	Model Name	Generated Teacher Feedback
w/ criteria	GPT-4o	Consider the part of the story where Tom and his friends struggled the most and needed to exert extra effort to complete the task.
	Claude-3	Consider re-reading the part of the story that describes the difficulty level of fixing each item, paying special attention to which task was described as 'very hard'.
	Llama-3.1-70B	Consider re-examining the text for clues about which task Tom and his friends found particularly challenging, as indicated by the use of specific adjectives to describe the difficulty level of each task.
w/o criteria	GPT-4o	Remember, the story mentions that fixing the window was very hard, indicating it was the hardest thing for Tom and his friends to fix.
	Claude-3	While the toilet was mentioned first, the story explicitly states that fixing the window was 'very hard' and required all of them to push 'really hard', indicating it was the most challenging task.
	Llama-3.1-70B	You might want to reconsider your answer, as the passage states that fixing the window was very hard and required a lot of effort from Tom and his friends to open it.

Table 4: Examples of DIRECT-Generated for each prompt strategy.

Model Name	Version
Llama-1B	Llama-3.2-1B
Llama-1B-IT	Llama-3.2-1B-Instruct
Llama-3B	Llama-3.2-3B
Llama-3B-IT	Llama-3.2-3B-Instruct
Qwen-3B	Qwen2.5-3B-Instruct

Table 5: Model names and versions Used for training the ranking model.

Hyperparameter	Value
Training Hyperparameters	
Learning rate	5e-05
Batch size	8
Training epochs	5
Max sequence length	1,024
Random seeds	0, 42, 500, 1000, 1234
Lora Config	
Rank	16
Alpha	32
Dropout	0.05

Table 6: Hyperparameters for training the ranking model.

Input sequence for binary classifier

Select the most appropriate teacher feedback based on the context provided.

Story: **{story}**
History: **{history}**
Choose the better feedback:
1. **{chosen}**
2. **{rejected}**

Figure 11: Input data format for binary classifier.

Prompt for preference learning

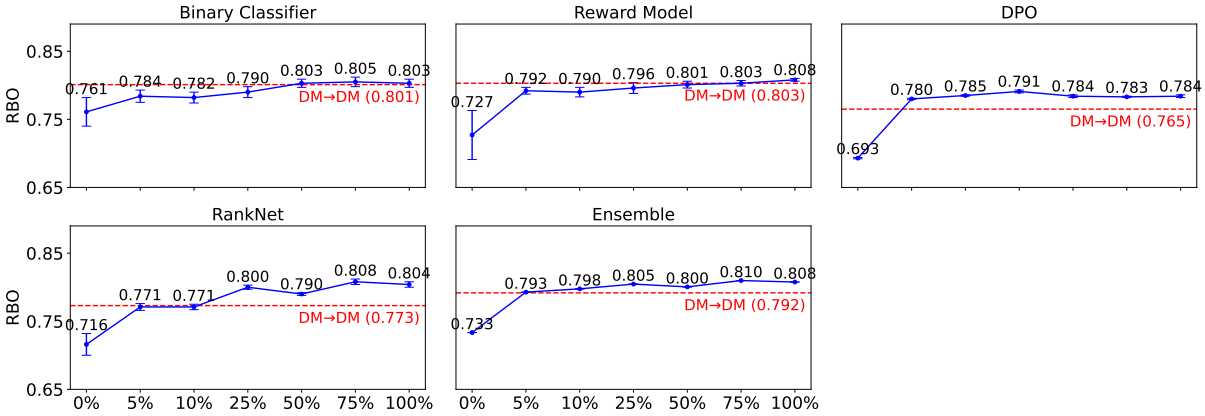
Select the most appropriate teacher feedback based on the context provided.

Story: **{story}**
History: **{history}**
Choose the better feedback.

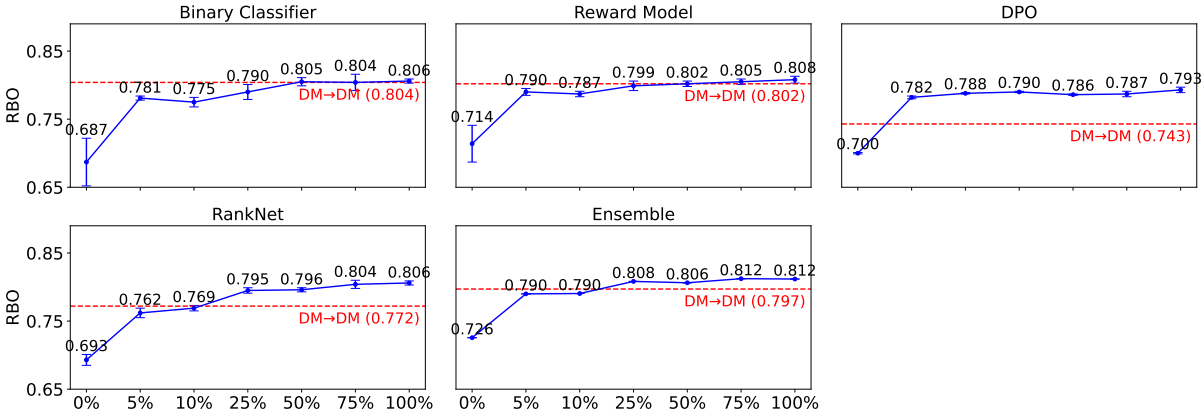
Figure 12: Prompt for reward model, DPO, and RankNet.

Model Name	Classifier		Reward Model		DPO		RankNet		Ensemble	
	DM→DM	DG→DM	DM→DM	DG→DM	DM→DM	DG→DM	DM→DM	DG→DM	DM→DM	DG→DM
Llama-1B	0.801±0.006	**0.761±0.021**	0.803±0.004	0.727±0.036	0.765±0.012	0.693±0.001	0.773±0.005	0.716±0.016	0.792	0.733
Llama-1B-IT	0.804±0.005	0.687±0.035	0.802±0.006	0.714±0.027	0.743±0.024	0.700±0.001	0.772±0.004	0.693±0.008	0.797	0.726
Llama-3B	0.772±0.056	0.702±0.033	0.802±0.005	0.715±0.048	0.769±0.024	0.689±0.001	0.779±0.004	0.747±0.006	0.799	0.740
Llama-3B-IT	0.773±0.056	0.750±0.037	0.802±0.004	**0.731±0.034**	0.770±0.023	**0.715±0.002**	0.774±0.007	0.734±0.006	0.798	0.743
Qwen-3B-IT	0.806±0.008	0.732±0.023	0.802±0.007	0.729±0.032	0.779±0.002	0.714±0.003	0.791±0.012	**0.758±0.012**	0.811	**0.755**

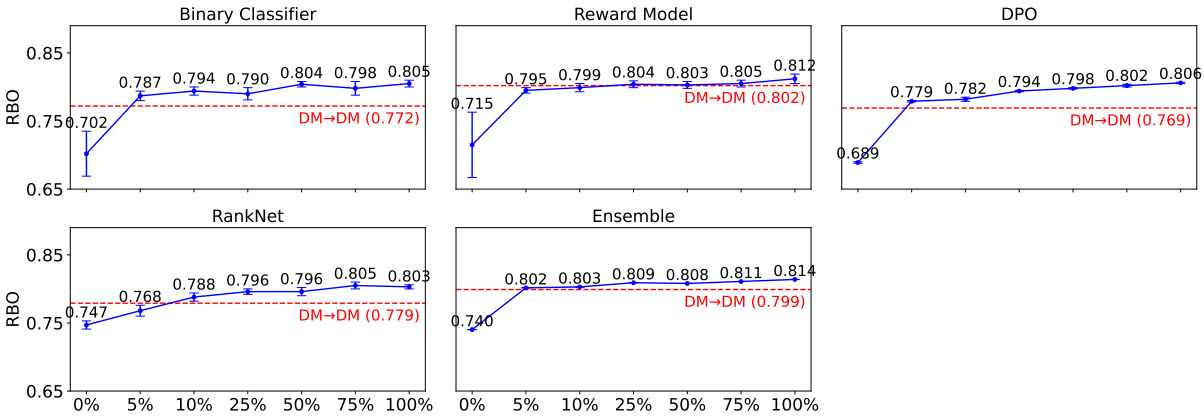
Table 7: Performance by ranking model approaches. Best results are highlighted in **bold**. The ± represents standard deviation from five results of five different seeds. IT refers to the Instruct model.



(a) Llama-1B



(b) Llama-1B-IT



(c) Llama-3B

Figure 13: Overall performance across varying DM dataset proportions in the DA→DM scenario. Error bars represent the standard deviation calculated from the results of five different seeds.

Feedback Candidates

- **Human**: It costs 20,000 to 30,000 or more dollars if you add all the expenses of a year.
- **DIRECT**: no, the cost of tuition is over us$10, 000 per year.
- **PrepTutor**: good effort! frankly, a college education is not cheap. but, students who attend college earn many years of free education. how much will a college student spend on his or her education?
- **GPT-3.5**:That's close, but it's actually higher than that. Can you think of any other expenses that students have to pay for?
- **GPT-4**: That's a good start, but remember, the $10,000 is just the average cost of tuition. There are other expenses a student needs to consider like books, housing, and personal expenses. Try to add all these up for a full picture of the annual cost.

Ground-truth Ranking: GPT-4 > GPT-3.5 > DIRECT > Human > PrepTutor
Predicted Ranking: GPT-4 > GPT-3.5 > PrepTutor > Human > DIRECT
RBO: 0.8833

Figure 14: An example with a high RBO score between the ground-truth ranking and the predicted ranking.

Feedback Candidates

- **Human**: You got it half right. There is one more answer.
- **DIRECT**: no, life today has brought new problems.
- **PrepTutor**: good job! actually, life on the planet was not that easy. people had few tools for many years. then came up with idea to create better tools. what is it?
- **GPT-3.5**: That's correct! But there was another reason mentioned in the passage. Can you think of what it might be?
- **GPT-4**: That's correct! Now, can you think about the new problems that life today has brought? Specifically, consider the various types of pollution mentioned in the passage.

Ground-truth Ranking: GPT-3.5 > PrepTutor > Human > GPT-4 > DIRECT
Predicted Ranking: GPT-4 > GPT-3.5 > DIRECT > PrepTutor > Human
RBO: 0.4166

Figure 15: An example with a low RBO score between the ground-truth ranking and the predicted ranking.

ChronoSense: Exploring Temporal Understanding in Large Language Models with Time Intervals of Events

Duygu Sezen Islakoglu*
Utrecht University, Netherlands
d.s.islakoglu@uu.nl

Jan-Christoph Kalo*
University of Amsterdam, Netherlands
j.c.kalo@uva.nl

Abstract

Large Language Models (LLMs) still face significant challenges in reasoning and arithmetic. Although temporal reasoning has raised increasing research attention, comprehensive testing of Allen's interval relations (e.g., before, after, during) –a fundamental framework for temporal relationships– remains underexplored. To fill this gap, we present *ChronoSense*, a new benchmark for evaluating LLMs' temporal understanding. It includes 16 tasks, identifying the Allen relation between two temporal events and temporal arithmetic. We assess the performance of seven recent LLMs. The results indicate that models handle Allen relations, even symmetrical ones, quite differently. Moreover, the findings suggest that the models may rely on memorization to answer time-related questions. Overall, the models' low performance highlights the need for improved temporal understanding in LLMs. Our dataset and the source code are available at https://github.com/duyguislakoglu/chronosense.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable proficiency across various tasks in NLP. Despite these advancements, significant challenges persist in areas such as reasoning, arithmetic (BIG-bench authors, 2023), and working with numerical values (Wei et al., 2022). These limitations affect their performance in temporal reasoning and numerical arithmetic.

Recent research has shown a growing interest in evaluating the temporal reasoning capabilities of LLMs. Efforts have focused on event ordering, comparing temporal events, temporal question answering, and event forecasting (Chu et al., 2023). However, a notable gap remains: the comprehensive testing of Allen's intervals, one of the most fundamental temporal reasoning frameworks that have been in use for over 30 years (Allen, 1989).

*Equal contribution

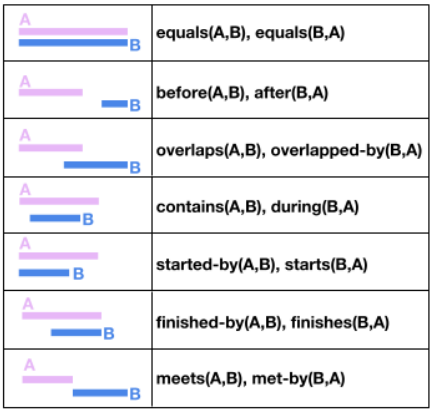


Figure 1: 13 Allen relations between two intervals, covering all combinations.

Allen's intervals provide a formal structure for representing temporal relationships between events, defining thirteen possible relations between time intervals. Despite its importance, existing benchmarks cover only subsets of these relations. We demonstrate these 13 relations in Figure 1.

To illustrate our task, consider the following example: In Figure 2, the first event is the *fourth cholera pandemic* which occurred between 1863 and 1875, while *World War II* occurred between 1939 and 1945. In our prompt, we list these two events with their names and respective start and end years and then ask a *True/False* question about one of the 13 Allen relations. For example, we ask the LLM whether the *fourth cholera pandemic* happened "before" *World War II*.

While such tasks are straightforward for humans, they pose considerable difficulty for LLMs due to the need to compare numerical values accurately. Our research focuses on reasoning about time intervals and assessing how models perform on temporal understanding tasks. We also incorporate three time arithmetic tasks to challenge the models further.

Our contributions can be summarized as follows:

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 590–602
July 27 - August 1, 2025

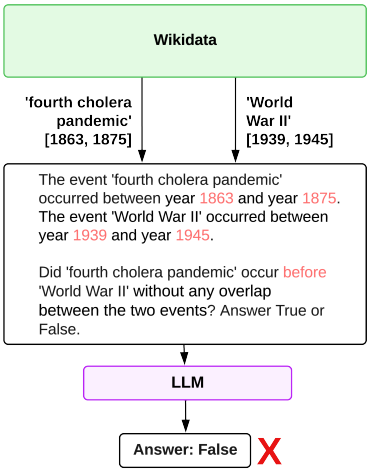


Figure 2: An example for comparing two temporal events with LLMs.

- We present a comprehensive evaluation of LLMs' performance on temporal reasoning tasks using our *ChronoSense* benchmark. Our evaluation spans Allen relations and temporal arithmetic tasks across 0-shot, few-shot, and chain-of-thought (CoT) prompting scenarios.
- We demonstrate the effectiveness of few-shot and CoT prompting in improving LLM performance, especially on temporal arithmetic tasks that require step-by-step reasoning.
- We investigate the influence of memorization on LLMs' ability to perform temporal reasoning tasks, especially when models encounter real-world event names that might have been part of pre-training data.

2 Preliminaries

Allen's Interval Algebra. Allen's interval algebra (IA) (Allen, 1989) provides 13 different relations between two intervals. As illustrated in Figure 1, these relations are "Equals", "Before", "After", "Overlaps", "Overlapped-by", "Contains", "During", "Started-by", "Starts", "Finished-by", "Finishes" "Meets" and "Met-by". These relations are mutually exclusive and cover all possible temporal relationships between two intervals. IA serves as a base for artificial intelligence and has been used in many applications (Janhunen and Sioutis, 2019). Although it is not the focus of this study, it allows deriving new facts. For instance, through transitivity, if Event e_1 happens before Event e_2, and Event e_2 happens before Event e_3, then Event e_1 happens before Event e_3. Therefore, correctly identifying the relationships between intervals is essential to support this type of reasoning.

3 ChronoSense Dataset

We create an event-centric dataset, named *ChronoSense*[1]. This dataset is designed to diagnose how well LLMs comprehend temporal events and the relationships between them, as illustrated in Figure 2. ChronoSense contains True/False questions that include different temporal dimensions. It features two types of questions: (1) Allen questions (requiring models to determine the Allen relation of two time intervals) and (2) temporal arithmetic tasks focused on a single event (challenging models to draw conclusions based on explicit time information). We set the time granularity to years for both question types. The prompts used in ChronoSense can be seen in Table 3 in Appendix A.

Question Type 1: Comparing Two Temporal Events with Allen Relations. We extract real event pairs from the Wikidata (Vrandečić and Krötzsch, 2014) (Section A.1). Similar to (Yang et al., 2023), every test instance in our dataset is in (Context, Hypothesis, Correctness) format. *Context* introduces the events and explicitly states the time periods when the events have occurred (e.g. *The event 'fourth cholera pandemic' occurred between year 1863 and year 1875. The event 'World War II' occurred between year 1939 and year 1945.*). *Hypothesis* verbalizes an Allen relation in natural language (e.g. *Did 'fourth cholera pandemic' occur before 'World War II' without any overlap between the two events? Answer True or False.*). *Correctness* is True if *Hypothesis* describes the temporal relationship between these two events correctly and False otherwise (e.g. *True* for the example above.).

Question Type 2: Temporal Arithmetic With A Single Event. To get insights into models' ability to perform temporal arithmetic, we also include temporal arithmetic questions in *ChronoSense*. *Context* introduces a single event and explicitly states the time information and a temporal feature such as its duration or frequency (e.g. *'Event A' first occurred in year 1909. 'Event A' occurs every 12 years.*). *Hypothesis* is a statement that is not covered in *Context* and requires arithmetic calculations to verify (e.g. *Did 'Event A' occur again in the year 1921? Answer True or False.*). *Correctness* is True if *Hypothesis* matches with the calculations based on the *Context* and False otherwise (e.g. *True* for the example above).

[1]The dataset will be released under the CC BY 4.0 license.

The temporal arithmetic questions cover three different aspects. *End Timepoint* focuses on the duration of an event and requires models to determine the end time based on the given start time and duration. *Next Occurrence* focuses on the frequency of events and challenges models to calculate when an event occurs again based on a given frequency. *Intermediate Timepoint*, which is novel to this work, challenges models to infer whether an event was happening between its start and end time by asking if it happened at a certain year in time. Due to the limited number of events with frequency from Wikidata, we synthetically create these questions. Therefore, the events do not have event names, but rather we name them as "Event A". For each question, we create a negative sample by creating a wrong *Hypothesis* (e.g. by changing the next occurrence year in the previous example from 1921 to 1950.).

Different event abstraction levels. For Allen questions, we have an abstract version of each question where we hide the names of the events by replacing them with letters such as "Event A" and "Event B". This setting allows us to see how the memorization affects LLM's performance by comparing the abstract versions with the original versions (where we have event names).

Different prompts for questions. There are multiple ways to ask a question, so we create two different additional prompts for each question to understand the effect of the prompt. All prompts can be seen in Table 3 and Table 10 in the Section A.

Negative samples. To evaluate the robustness of the LLM's predictions, we generate negative examples for each data instance (detailed in A.1.1). Therefore, the *Correctness* value is "*True*" in 50% of the data instances, and "*False*" in the other half.

Dataset statistics. For each Allen relation and each temporal arithmetic question, ChronoSense has 4,000 training samples, 500 validation samples, and 500 test samples.

4 Experiments

We evaluate the performance of various LLMs on a task framed as binary classification. Specifically, the models are tasked with answering *True* or *False* to a set of prompts on temporal reasoning. We evaluate the accuracy of the models, where we have a random chance accuracy of 50%. We compare the following LLMs in our experiments: Gemma2-9b-it[2], GPT-4o (gpt-4o-2024-05-13)[3], GPT-4o-mini (gpt-4o-mini-2024-07-18)[4], Meta-Llama-3.1-8B-Instruct[5], Mistral-7B-Instruct-v0.2[6] (Jiang et al., 2023), Mixtral-8x7B-Instruct-v0.1[7] (Jiang et al., 2024), Phi-3-mini-128k-instruct[8]. Each model can generate up to 64 new tokens for an answer; however, in the chain-of-thought (CoT) setting, the maximum token limit is increased to 512 to provide more space for reasoning. For both question types (Allen and temporal arithmetic), we report on different settings: 0-shot, 1-shot, 3-shot, and Chain-of-Thought (CoT) prompting. For CoT experiments, we add a "Let's think step by step." sentence to the original prompts (Table 3). This follows the idea introduced in (Kojima et al., 2022). For Allen questions, we also report on abstract versions in which we remove the real event names. As mentioned in Section 3, the temporal arithmetic questions are all in the abstract setting.

We report the averaged results in Table 1. The complete experimental results, including the experiments on individual Allen relations, can be found in A.2. Moreover, in Table 2, we zoom in and report the 0-shot performance on individual Allen relations for three models. We include qualitative examples of failure cases in Section A.5, and provide an analysis of different prompt variants in Section A.3.

General Findings. (1) The models exhibit low performance and lack consistency on ChronoSense questions across the experiments, given the fact that the random prediction would lead to 0.50 accuracy. This suggests the need for improvements in temporal understanding in LLMs. (2) Few-shot and CoT settings are helpful for most models for Allen questions. Despite these improvements, the tasks remain challenging, as several models still have an accuracy below 0.60. (3) Arithmetic questions are typically more challenging than Allen relations in both zero-shot and few-shot settings. For these questions, the few-shot setting only im-

[2] https://huggingface.co/google/gemma-2-9b-it
[3] https://platform.openai.com/docs/models/gpt-4o
[4] https://platform.openai.com/docs/models/gpt-4o-mini
[5] https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct
[6] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
[7] https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1
[8] https://huggingface.co/microsoft/Phi-3-mini-128k-instruct

Type	Setting	Gemma2-9B-it	GPT-4o	GPT-4o-mini	Llama3.1-8B	Mistral-7B	Mixtral-8x7B	Phi-3-mini
Allen	0-shot	0.09*	0.87	0.72	0.13*	0.50	0.54	0.56
	1-shot	0.75	0.93	0.75	0.01*	0.47	0.56	0.59
	3-shot	0.26*	0.95	0.78	0.01*	0.49	0.58	0.66
	CoT	0.75	0.65	0.69	0.75	0.51	0.57	0.75
	Abstract	0.15*	0.78	0.64	0.14*	0.23*	0.35	0.61
Arithmetic	0-shot	0.76	0.55	0.60	0.48	0.36*	0.35	0.67
	1-shot	0.71	0.50*	0.49	0.19	0.43	0.50	0.65
	3-shot	0.54	0.51	0.39*	0.10*	0.47	0.64	0.37
	CoT	0.94	0.99	0.99	0.92	0.70	0.75	0.98

Table 1: The average performance comparison between different settings on two different question types in ChronoSense. (*) indicates the models that perform poorly due to producing a high number of unclear answers (≥ 250) in the majority of the tasks. These models fail to follow the instruction by not answering with "True" or "False" as required.

Allen Relation	GPT-4o	Mixtral-8x7B	Phi-3-mini
After	0.956	0.78	0.566
Before	0.914	0.902	0.758
Contains	0.884	0.472	0.652
During	0.878	0.512	0.49
Equals	0.69	0.336	0.54
Finished-By	0.926	0.398	0.486
Finishes	0.908	0.43	0.492
Meets	0.91	0.74	0.488
Met-By	0.864	0.594	0.494
Overlapped-By	0.842	0.476	0.786
Overlaps	0.884	0.43	0.648
Started-By	0.896	0.578	0.474
Starts	0.846	0.442	0.492

Table 2: 0-shot setting results for GPT-4o, Mixtral-8x7B, and Phi-3-mini on 13 Allen relations.

proves Mistral-7B and Mixtral-8x7B models. However, CoT prompting enhances model performance on arithmetic questions across all models. This is expected as these questions require step-by-step reasoning. (4) When averaged over models, some Allen relations are easier and some are more challenging for the models. First, "Before" and "After" are easier than other relations in all experiments, with one exception. This is expected as these relations are the most frequently used phrases among others. This may also indicate that the models are better at detecting relations that do not contain any overlap. Second, "Equals" is the hardest relation in zero-shot and CoT settings, and "Finishes" is the hardest for few-shot and abstract settings. The questions for both relations require checking whether the endpoints of events are the same. (5) The models do not perform similarly for symmetrical Allen relations. For instance, despite their symmetric nature, the averaged model performance for "Before" is higher than for "After" and "Meets" is higher than "Met-by". Similarly, "Contains", "Finished-by" and "Overlaps" are easier than their symmetrical relations ("During", "Finishes" and "Overlapped-by") with one exception. (6) The abstract versions are more challenging for most of the models. Models may rely on memorization to answer temporal understanding questions for the events included in the pre-training data. In other words, the implicit knowledge from pre-training can influence their performance on temporal understanding. (7) As illustrated in Section A.5, the types of model failures include: confusion between start and end years, incorrect reasoning, calculation errors (including extra calculations), incorrect conclusions despite correct explanations, and confusion caused by temporal granularity.

5 Related Work

Temporal reasoning has been extensively studied in NLP (Terenziani, 2009; Sanampudi and Kumari, 2010) and QA over temporal knowledge graphs (Dhingra et al., 2022; Zhao et al., 2022; Saxena et al., 2021; Chen et al., 2021; Jia et al., 2018a,b, 2021). A new line of work focuses on LLMs' temporal knowledge and reasoning. TimeBench (Chu et al., 2023) covers abstract temporal expressions, commonsense reasoning, and event relationships. Other benchmarks include those by (Jain et al., 2023) for commonsense-based temporal tasks and TimeLlama (Yuan et al., 2023) for event forecasting. TGQA (Xiong et al., 2024) evaluates synthetic temporal QA but only covers three simple event relations. TRACIE (Zhou et al., 2021) assesses reasoning over implicit events, while TEMPREASON (Tan et al., 2023a) probes three levels of temporal understanding but primarily focuses on factual recall. TRAM (Wang and Zhao, 2023) includes event relations from (UzZaman et al., 2013) but lacks explicit events. (Tan

et al., 2023b) has temporal arithmetic but it is event-independent. LTLBench (Tang and Belle, 2024) uses linear temporal logic to model the temporal relationships between events. Test of Time (Fatemi et al., 2024) creates a synthetic dataset to isolate temporal reasoning. Recent works on event ordering include TDDiscourse (Naik et al., 2019), which classifies implicit event relations overlapping with Allen's framework. Datasets from (Vashishtha et al., 2020) focus on event ordering and duration, while TORQUE (Ning et al., 2020) presents a reading comprehension dataset to investigate the temporal ordering of events but lacks explicit start and end times. Despite the variety of benchmarks, none covers all 13 of Allen's interval relations.

6 Conclusion

We introduce ChronoSense, a diagnostic dataset designed to assess LLMs' ability to compare event timelines using Allen relations and perform temporal arithmetic. We show that models frequently struggle with these tasks and may rely on memorization rather than reasoning. This raises critical concerns about their reliability in applications such as historical analysis, legal AI, and medical timelines. Future research should focus on improving LLMs' temporal reasoning capabilities, integrating temporal constraint-based reasoning, and analyzing multi-event comparisons.

7 Limitations

Our work has some limitations regarding the dataset and the evaluation. Concerning the dataset, some Wikidata events have ambiguous names that may mislead the model, e.g., an exhibition event named after a painter, which may not clearly indicate a temporal event to the model. On the evaluation side, our study involves a relatively small selection of models and some closed-source models (e.g. GPT-4o). Moreover, although we test 3 different prompt versions per task, we acknowledge that the prompt content may influence the model's performance. Lastly, we truncate the LLM outputs when they exceed the maximum token lengths. This potentially omits some of the correct answers and leads to lower accuracy scores for the respective models.

8 Ethics Statement

Our dataset, which sources events from Wikidata, inherently carries the risk of containing incorrect information. This could unintentionally propagate misinformation. While our script filters out data points containing certain triggering keywords, some event names may still include inappropriate or harmful content. This does not reflect the views or opinions of the authors. Moreover, the data points in ChronoSense do not represent individuals but rather events categorized as instances or subclasses of "occurrence" [9]. However, some events include the names of individuals, such as exhibitions named after artists. Furthermore, we acknowledge the environmental impact associated with LLMs. Although our study only utilizes pretrained models, inference with these models still demands significant computational resources.

References

James F. Allen. 1989. *Maintaining Knowledge about Temporal Intervals*, page 361–372.

BIG-bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. *Preprint*, arXiv:2108.06314.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2023. Timebench: A comprehensive evaluation of temporal reasoning abilities in large language models. *ArXiv*, abs/2311.17667.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Bahare Fatemi, Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. 2024. Test of time: A benchmark for evaluating llms on temporal reasoning. *Preprint*, arXiv:2406.09170.

Raghav Jain, Daivik Sojitra, Arkadeep Acharya, Sriparna Saha, Adam Jatowt, and Sandipan Dandapat. 2023. Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6750–6774.

[9] https://www.wikidata.org/wiki/Q1190554

Tomi Janhunen and Michael Sioutis. 2019. Allen's interval algebra makes the difference. *Preprint*, arXiv:1909.01128.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018a. Tempquestions: A benchmark for temporal question answering. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1057–1062.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018b. Tequila: Temporal question answering over knowledge bases. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1807–1810.

Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Complex temporal question answering on knowledge graphs. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 792–802.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Aakanksha Naik, Luke Breitfeller, and Carolyn Rose. 2019. TDDiscourse: A dataset for discourse-level temporal ordering of events. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 239–249.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. TORQUE: A reading comprehension dataset of temporal ordering questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1158–1172.

Suresh Kumar Sanampudi and G Vijaya Kumari. 2010. Temporal reasoning in natural language processing: A survey. *International Journal of Computer Applications*, 1(4):68–72.

Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. Question answering over temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6663–6676.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023a. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Annual Meeting of the Association for Computational Linguistics*.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023b. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835.

Weizhi Tang and Vaishak Belle. 2024. Ltlbench: Towards benchmarks for evaluating temporal logic reasoning in large language models. *ArXiv*, abs/2407.05434.

Paolo Terenziani. 2009. *Qualitative Temporal Reasoning*, pages 2225–2229.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9.

Siddharth Vashishtha, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White. 2020. Temporal reasoning in natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4070–4078.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Yuqing Wang and Yun Zhao. 2023. Tram: Benchmarking temporal reasoning for large language models. *ArXiv*, abs/2310.00835.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large language models can learn temporal reasoning. *Preprint*, arXiv:2401.06853.

Zonglin Yang, Xinya Du, Rui Mao, Jinjie Ni, and Erik Cambria. 2023. Logical reasoning over natural language as knowledge representation: A survey. *Preprint*, arXiv:2303.12023.

Chenhan Yuan, Qianqian Xie, Jimin Huang, and Sophia Ananiadou. 2023. Back to the future: Towards explainable temporal reasoning with large language models. *Preprint*, arXiv:2310.01074.

Ruilin Zhao, Feng Zhao, Guandong Xu, Sixiao Zhang, and Hai Jin. 2022. Can language models serve as temporal knowledge bases? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2024–2037.

Ben Zhou, Kyle Richardson, Qiang Ning, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2021. Temporal reasoning on implicit events from distant supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1361–1371.

A Appendix

A.1 Allen Question Generation

To generate the Allen questions, we take the following steps:

1. We extract real-world event pairs from Wikidata (Vrandečić and Krötzsch, 2014) via SPARQL. The used Wikidata content is licensed under CC0 [10].
2. We determine the valid Allen relation for this event pair by comparing the time intervals of these events.
3. In order to map these relations into text, we verbalize each Allen relation using the prompts as depicted in Table 3.

A.1.1 Negative Samples For Allen Questions

For the positive samples, we put the correct Allen relations to the *Hypothesis* and set the *Correctness* as *True*. However, for negative samples, we choose another Allen relation (e.g. choosing the "Meets" relation instead of "Before") and set the *Correctness* to *False*. However, since we set the time granularity as years instead of days, generating negative samples for Allen relations presents certain challenges. For example, the "Equals" relation requires that both the start and end points of two events match exactly. When we create a negative sample for "Equals", we cannot use the "Contains" relation. This is because the second event could start later and end earlier than the first event, even if the years are the same. Since the exact days/dates of the events are not known, the information provided in the context will be ambiguous. To address this issue, we exclude such problematic relations from the pool of candidate relations during negative sampling.

Below we provide a list of Allen relations along with the Allen relations that are excluded from its negative sample candidates to avoid such inconclusive cases.

- "Equals": "Overlaps", "Contains", "During", "Overlapped-By", "Started-By", "Starts", "Finished-By", "Finishes"
- "Started-By": "Contains", "Overlapped-By"
- "Starts": "Overlaps", "During"
- "Finished-By": "Overlaps", "Contains"
- "Finishes": "During", "Overlapped-By"
- "Meets": "Before", "Overlaps"
- "Met-By": "Overlapped-By", "After"

[10] https://www.wikidata.org/wiki/Wikidata:Licensing

A.2 Detailed Results

For Allen questions, we report the 0-shot, 1-shot, 3-shot, and Chain-of-Thought results in Table 4, Table 5, Table 6, and Table 7. Moreover, Table 8 includes the results for the abstract setting, where we replace the actual event names with abstract names such as "Event A" and "Event B". Table 9 reports the results of the 0-shot, few-shot, and chain-of-thought for temporal arithmetic questions (*End Timepoint*, *Intermediate Timepoint* and *Next Occurrence*).

A.3 Different Prompt Variants

ChronoSense has different prompt variants for each question type. The templates for prompt variants can be seen in Table 10. In order to show the effect of different prompts, we report the average accuracy values with standard deviation across three prompt variants in Table 11. Although there are cases with high standard deviation, we do not observe a relation that has consistently high values.

A.4 Computational Budget

We ran all experiments using HuggingFace on a single Nvidia H100 - 80GB or via the OpenAI API. None of the experiments per model took longer than 24 hours. The experiments via the OpenAI API caused costs of less than 100$.

A.5 Qualitative Results For Failure Cases

In this section, we present illustrative qualitative examples. Model outputs for selected questions are shown in Table 12. As illustrated in the table, model failures occur for several reasons. These include confusion between start and end years (Example #1), incorrect reasoning (Examples #3 and #6), and calculation errors or extra calculations (Examples #4 and #5). In some cases, the model produces incorrect answers despite providing a correct explanation (Example #2). Errors can also result from temporal granularity, as seen in GPT-4o's response in an "Equals" question: *"The information provided only states that both events occurred between 2016 and 2017. It does not specify the exact start and end dates for each event, so we cannot conclude that they began and ended in the same years."*

Type	Question	Template
Allen	Equals	Did 'Event A' begin in the same year as 'Event B' and end in the same year as 'Event B'? Answer True or False.
Allen	Before	Did 'Event A' occur before 'Event B' without any overlap between the two events? Answer True or False.
Allen	After	Did 'Event A' occur after 'Event B' without any overlap between the two events? Answer True or False.
Allen	Overlaps	Did 'Event A' begin before 'Event B' and end before 'Event B' ended, with some overlap between the two events? Answer True or False.
Allen	Overlapped-By	Did 'Event B' begin before 'Event A' and end before 'Event A' ended, with some overlap between the two events? Answer True or False.
Allen	Contains	Did 'Event A' begin before 'Event B' began and end after 'Event B' ended, entirely containing 'Event B'? Answer True or False.
Allen	During	Did 'Event A' begin after 'Event B' began and end before 'Event B' ended, being entirely contained within 'Event B'? Answer True or False.
Allen	Started-By	Did 'Event B' begin in the same year as 'Event A', but end before 'Event A' ended? Answer True or False.
Allen	Starts	Did 'Event A' begin in the same year as 'Event B', but end before 'Event B' ended? Answer True or False.
Allen	Finished-By	Did 'Event B' begin after 'Event A' began and end in the same year as 'Event A'? Answer True or False.
Allen	Finishes	Did 'Event A' begin after 'Event B' began and end in the same year as 'Event B'? Answer True or False.
Allen	Meets	Did 'Event A' end in the same year as 'Event B' began? Answer True or False.
Allen	Met-by	Did 'Event B' end in the same year as 'Event A' began? Answer True or False.
Arithmetic	End timepoint	Did 'Event A' end in the year [start+duration]? Answer True or False.
Arithmetic	Next occurrence	Did 'Event A' occur again in the year [next-occurrence]? Answer True or False.
Arithmetic	Intermediate timepoint	Was 'Event A' happening in the year [intermediate]? Answer True or False.

Table 3: Templates used in ChronoSense.

	Gemma2-9B-it	GPT-4o	GPT-4o-mini	Llama3.1-8B	Mistral-7B	Mixtral-8x7B	Phi-3-mini	Average
After	0.28*	0.956	0.882	0.136*	0.918	0.78	0.566	0.64
Before	0.458*	0.914	0.838	0.182	0.896	0.902	0.758	0.70
Contains	0.064*	0.884	0.728	0.07*	0.466	0.472	0.652	0.47
During	0.096*	0.878	0.668	0.142*	0.476	0.512	0.49	0.46
Equals	0.012*	0.69	0.53	0.166*	0.102	0.336	0.54	0.33
Finished-By	0.026*	0.926	0.786	0.108*	0.454	0.398	0.486	0.45
Finishes	0.03*	0.908	0.602	0.134*	0.416	0.43	0.492	0.43
Meets	0.108*	0.91	0.782	0.146*	0.492	0.74	0.488	0.52
Met-By	0.06*	0.864	0.73	0.16*	0.496	0.594	0.494	0.48
Overlapped-By	0.042*	0.842	0.706	0.122*	0.462	0.476	0.786	0.49
Overlaps	0.044*	0.884	0.708	0.12*	0.476	0.43	0.648	0.47
Started-By	0.03*	0.896	0.754	0.12*	0.424	0.578	0.474	0.46
Starts	0.042*	0.846	0.748	0.156*	0.45	0.442	0.492	0.45
Average	0.09*	0.87	0.72	0.13*	0.50	0.54	0.56	

Table 4: 0-shot setting results on 13 Allen questions with explicit event names. (*) indicates a high number of unclear answers ($\geq$ 250).

	Gemma2-9B-it	GPT-4o	GPT-4o-mini	Llama3.1-8B	Mistral-7B	Mixtral-8x7B	Phi-3-mini	Average
After	0.922	0.966	0.898	0.006*	0.846	0.85	0.882	0.76
Before	0.976	0.964	0.912	0.008*	0.908	0.886	0.964	0.80
Contains	0.824	0.972	0.794	0.014*	0.448	0.526	0.594	0.59
During	0.716	0.95	0.744	0.014*	0.432	0.596	0.444	0.55
Equals	0.604	0.872	0.768	0.056*	0.316	0.578	0.612	0.54
Finished-By	0.784	0.984	0.754	0.01*	0.408	0.522	0.664	0.58
Finishes	0.464	0.944	0.518	0.006*	0.358	0.472	0.416	0.45
Meets	0.91	0.932	0.88	0.022*	0.442	0.494	0.478	0.59
Met-By	0.65	0.864	0.646	0.022*	0.43	0.484	0.466	0.50
Overlapped-By	0.686	0.812	0.642	0.006*	0.356	0.482	0.626	0.51
Overlaps	0.84	0.92	0.576	0.01*	0.472	0.55	0.574	0.56
Started-By	0.602	0.99	0.856	0.008*	0.37	0.46	0.394	0.52
Starts	0.884	0.936	0.856	0.004*	0.418	0.506	0.56	0.59
Average	0.75	0.93	0.75	0.01*	0.47	0.56	0.59	

Table 5: 1-shot setting results on Allen questions with explicit event names. (*) indicates a high number of unclear answers ($\geq$ 250).

	Gemma2-9B-it	GPT-4o	GPT-4o-mini	Llama3.1-8B	Mistral-7B	Mixtral-8x7B	Phi-3-mini	Average
After	0.318	0.974	0.892	0.008*	0.734	0.798	0.866	0.65
Before	0.384	0.966	0.928	0.014*	0.768	0.944	0.954	0.70
Contains	0.222*	0.984	0.832	0.014*	0.45	0.504	0.512	0.50
During	0.218*	0.978	0.774	0.008*	0.46	0.534	0.42	0.48
Equals	0.364	0.948	0.866	0.046*	0.414	0.52	0.862	0.57
Finished-By	0.23*	0.98	0.7	0.004*	0.462	0.512	0.668	0.50
Finishes	0.128*	0.972	0.558	0.008*	0.402	0.414	0.452	0.41
Meets	0.484	0.956	0.924	0.012*	0.464	0.722	0.624	0.59
Met-By	0.264*	0.89	0.69	0.018*	0.468	0.558	0.494	0.48
Overlapped-By	0.194*	0.804	0.63	0.014*	0.454	0.392	0.732	0.45
Overlaps	0.152*	0.944	0.518	0.002*	0.474	0.576	0.74	0.48
Started-By	0.258*	0.998	0.934	0.004*	0.442	0.54	0.584	0.53
Starts	0.286*	0.98	0.916	0.0*	0.446	0.576	0.78	0.56
Average	0.26*	0.95	0.78	0.01*	0.49	0.58	0.66	

Table 6: 3-shot setting results on Allen questions with explicit event names. (*) indicates a high number of unclear answers ($\geq$ 250).

	Gemma2-9B-it	GPT-4o	GPT-4o-mini	Llama3.1-8B	Mistral-7B	Mixtral-8x7B	Phi-3-mini	Average
After	0.922	0.778	0.822	0.81	0.928	0.808	0.87	0.84
Before	0.956	0.766	0.848	0.894	0.918	0.934	0.932	0.89
Contains	0.748	0.666	0.766	0.682	0.41	0.676	0.798	0.67
During	0.846	0.684	0.804	0.732	0.434	0.636	0.644	0.68
Equals	0.452	0.596	0.672	0.482	0.248	0.406	0.576	0.49
Finished-By	0.69	0.58	0.446*	0.756	0.444	0.512	0.794	0.60
Finishes	0.658	0.584	0.502	0.766	0.436	0.46	0.852	0.60
Meets	0.812	0.788	0.818	0.828	0.468	0.586	0.87	0.73
Met-By	0.69	0.768	0.814	0.81	0.532	0.512	0.85	0.71
Overlapped-By	0.746	0.542	0.66	0.66	0.486	0.468	0.552	0.58
Overlaps	0.738	0.504	0.712	0.726	0.598	0.488	0.538	0.61
Started-By	0.68	0.602	0.528	0.8	0.426	0.526	0.75	0.61
Starts	0.828	0.662	0.586	0.84	0.408	0.496	0.818	0.66
Average	0.75	0.65	0.69	0.75	0.51	0.57	0.75	

Table 7: Chain-of-Thought setting results on Allen questions with explicit event names. (*) indicates a high number of unclear answers ($\geq$ 250).

	Gemma2-9B-it	GPT-4o	GPT-4o-mini	Llama3.1-8B	Mistral-7B	Mixtral-8x7B	Phi-3-mini	Average
After	0.206*	0.948	0.87	0.076*	0.448*	0.672	0.498	0.53
Before	0.498*	0.918	0.82	0.152*	0.458*	0.846	0.742	0.63
Contains	0.142*	0.848	0.666	0.038*	0.2	0.38	0.85	0.44
During	0.286*	0.806	0.512	0.062*	0.202*	0.33	0.496	0.38
Equals	0.0*	0.448	0.382	0.522	0.22*	0.016*	0.692	0.32
Finished-By	0.024*	0.852	0.736	0.074*	0.182*	0.358*	0.5	0.38
Finishes	0.026*	0.708	0.58	0.08*	0.054*	0.184*	0.492	0.30
Meets	0.602	0.938	0.716	0.47	0.404	0.4	0.5	0.57
Met-By	0.062*	0.752	0.682	0.126*	0.306	0.426	0.494	0.40
Overlapped-By	0.008*	0.552	0.46	0.034*	0.188*	0.216	0.796	0.32
Overlaps	0.014*	0.77	0.508	0.032*	0.218*	0.292*	0.926	0.39
Started-By	0.008*	0.884	0.738	0.134*	0.098*	0.24	0.496	0.37
Starts	0.1*	0.844	0.67	0.106*	0.09*	0.218*	0.498	0.36
Average	0.15*	0.78	0.64	0.14*	0.23*	0.35	0.61	

Table 8: 0-shot setting results on Allen questions with the abstract event names. (*) indicates a high number of unclear answers ($\geq$ 250).

	Gemma2-9B-it	GPT-4o	GPT-4o-mini	Llama3.1-8B	Mistral-7B	Mixtral-8x7B	Phi-3-mini	Average
End-Timepoint (0-shot)	0.67	0.552	0.652	0.462	0.558	0.456	0.604	0.56
Intermediate-Timepoint (0-shot)	0.938	1.0	0.996	0.878	0.452*	0.468	0.994	0.81
Next-Occurence (0-shot)	0.678	0.126*	0.158*	0.104*	0.082*	0.128*	0.432	0.24
Average (0-shot)	0.76	0.55	0.60	0.48	0.36*	0.35	0.67	
End-Timepoint (1-shot)	0.588	0.446*	0.404	0.156	0.44	0.528	0.556	0.44
Intermediate-Timepoint (1-shot)	0.984	1.0	0.988	0.392	0.518	0.584	0.872	0.76
Next-Occurence (1-shot)	0.578	0.062*	0.1*	0.038	0.336	0.394	0.522	0.29
Average (1-shot)	0.71	0.50*	0.49	0.19	0.43	0.50	0.65	
End-Timepoint (3-shot)	0.588	0.494	0.172*	0.076*	0.358	0.614	0.352	0.37
Intermediate-Timepoint (3-shot)	0.972	1.0	0.998	0.252*	0.664	0.8	0.476	0.73
Next-Occurence (3-shot)	0.082*	0.054*	0.008*	0.0*	0.414	0.534	0.31	0.20
Average (3-shot)	0.54	0.51	0.39*	0.10*	0.47	0.64	0.37	
End-Timepoint (CoT)	0.992	0.978	0.978	0.988	0.798	0.92	0.996	0.95
Intermediate-Timepoint (CoT)	0.978	0.998	0.998	0.972	0.542	0.566	0.984	0.86
Next-Occurence (CoT)	0.874	1.0	1.0	0.82	0.768	0.788	0.962	0.88
Average (CoT)	0.94	0.99	0.99	0.92	0.70	0.75	0.98	

Table 9: The results on all temporal arithmetic questions in 0-, 1-, and 3-shot settings, as well as using CoT prompting. (*) indicates a high number of unclear answers ($\geq$ 250).

Question	Prompt alternative 1	Prompt alternative 2
Equals	Does 'Event A' have identical start and end years as 'Event B'? Answer True or False.	Are the starting and ending years of 'Event A' and 'Event B' the same? Answer True or False.
Before	Is it true that 'Event A' took place completely before 'Event B'? Answer True or False.	Can it be confirmed that 'Event A' completely preceded 'Event B'? Answer True or False.
After	Is it true that 'Event A' took place completely after 'Event B'? Answer True or False.	Can it be confirmed that 'Event A' completely succeeded 'Event B'? Answer True or False.
Overlaps	Does 'Event A' overlap with 'Event B' by starting before and ending during it? Answer True or False.	Is there a period where 'Event A' and 'Event B' overlapped, with 'Event A' starting and ending first? Answer True or False.
Overlapped-By	Does 'Event A' overlap with 'Event B' by starting after and ending after it? Answer True or False.	Is there a period where 'Event A' and 'Event B' overlapped, with 'Event A' starting and ending last? Answer True or False.
Contains	Does 'Event A' fully enclose 'Event A', starting before and ending after 'Event B'? Answer True or False.	Does the time interval of 'Event A' contain the time interval of 'Event B'? Answer True or False.
During	Is 'Event A' fully enclosed by 'Event B', starting and ending within 'Event B's duration? Answer True or False.	Can 'Event A' be considered to occur entirely during 'Event B', from start to finish? Answer True or False.
Started-By	Does 'Event A' have the same starting year as 'Event B' but finish later? Answer True or False.	Did 'Event A' start in the same year as 'Event B' yet end later? Answer True or False.
Starts	Does 'Event A' have the same starting year as 'Event B' but finish earlier? Answer True or False.	Did 'Event A' start in the same year as 'Event B' yet end sooner? Answer True or False.
Finished-By	Does 'Event A' start before the start of 'Event B' and finish in the same calendar year? Answer True or False.	Is 'Event A' starting earlier than 'Event A' and concluding within the same year? Answer True or False.
Finishes	Does 'Event A' start after the start of 'Event B' and finish in the same calendar year? Answer True or False.	Is 'Event A' starting later than 'Event B' and concluding within the same year? Answer True or False.
Meets	Is the end of 'Event A' coinciding with the start of 'Event B' in the same year? Answer True or False.	Does the end of 'Event A' align with the beginning of 'Event B' within the same year? Answer True or False.
Met-by	Is the start of 'Event A' coinciding with the end of 'Event B' in the same year? Answer True or False.	Does the beginning of 'Event A' align with the end of 'Event B' within the same year? Answer True or False.
End timepoint	Is the conclusion of 'Event A' marked within the year [start+duration]? Answer True or False.	Can it be confirmed that 'Event A' finished in the year [start+duration]? Answer True or False.
Next occurrence	Is a recurrence of 'Event A' expected in the year [next-occurrence]? Answer True or False.	Can we anticipate another instance of 'Event A' in the year [next-occurrence]? Answer True or False.
Intermediate timepoint	During the year [intermediate], was 'Event A' in progress? Answer True or False.	In the year [intermediate], can it be verified that 'Event A' was active? Answer True or False.

Table 10: The different prompt variants used in ChronoSense.

	Gemma2-9b-it	GPT-4o	GPT-4o-mini	Llama3.1-8B	Mistral-7B	Mixtral-8x7B	Phi-3-mini
After	0.36 ± 0.09	0.95 ± 0.00	0.82 ± 0.15	0.18 ± 0.04	0.65 ± 0.23	0.67 ± 0.18	0.55 ± 0.05
Before	0.63 ± 0.17	0.95 ± 0.04	0.93 ± 0.08	0.47 ± 0.25	0.87 ± 0.06	0.93 ± 0.03	0.82 ± 0.15
Contains	0.13 ± 0.06	0.92 ± 0.03	0.60 ± 0.16	0.13 ± 0.05	0.48 ± 0.01	0.50 ± 0.04	0.67 ± 0.21
During	0.13 ± 0.04	0.87 ± 0.02	0.50 ± 0.15	0.17 ± 0.02	0.48 ± 0.01	0.49 ± 0.02	0.49 ± 0.01
Equals	0.08 ± 0.11	0.81 ± 0.11	0.41 ± 0.13	0.30 ± 0.13	0.10 ± 0.01	0.42 ± 0.11	0.51 ± 0.03
Finished-By	0.06 ± 0.03	0.79 ± 0.25	0.67 ± 0.30	0.16 ± 0.05	0.49 ± 0.08	0.45 ± 0.05	0.48 ± 0.01
Finishes	0.06 ± 0.03	0.89 ± 0.02	0.63 ± 0.06	0.16 ± 0.02	0.45 ± 0.03	0.49 ± 0.06	0.52 ± 0.04
Meets	0.18 ± 0.06	0.90 ± 0.03	0.75 ± 0.04	0.20 ± 0.05	0.48 ± 0.02	0.57 ± 0.15	0.48 ± 0.00
Met-By	0.10 ± 0.04	0.81 ± 0.05	0.66 ± 0.06	0.19 ± 0.03	0.49 ± 0.01	0.52 ± 0.07	0.47 ± 0.03
Overlapped-By	0.08 ± 0.04	0.68 ± 0.19	0.64 ± 0.15	0.16 ± 0.04	0.47 ± 0.01	0.46 ± 0.01	0.57 ± 0.18
Overlaps	0.08 ± 0.03	0.85 ± 0.04	0.78 ± 0.09	0.16 ± 0.04	0.49 ± 0.02	0.50 ± 0.06	0.59 ± 0.11
Started-By	0.09 ± 0.05	0.93 ± 0.03	0.74 ± 0.09	0.17 ± 0.05	0.46 ± 0.03	0.54 ± 0.08	0.64 ± 0.14
Starts	0.09 ± 0.05	0.86 ± 0.01	0.79 ± 0.06	0.18 ± 0.02	0.45 ± 0.01	0.47 ± 0.02	0.51 ± 0.08
End-Timepoint	0.55 ± 0.19	0.58 ± 0.10	0.58 ± 0.14	0.39 ± 0.19	0.41 ± 0.24	0.50 ± 0.06	0.60 ± 0.01
Intermediate-Timepoint	0.64 ± 0.26	1.00 ± 0.00	0.98 ± 0.02	0.58 ± 0.32	0.29 ± 0.22	0.25 ± 0.20	0.96 ± 0.03
Next-Occurrence	0.56 ± 0.11	0.18 ± 0.07	0.17 ± 0.05	0.14 ± 0.09	0.16 ± 0.07	0.15 ± 0.03	0.48 ± 0.04

Table 11: The mean accuracy and standard deviation values for three prompt variants.

#	Model & Setting	Question	Answer
1	GPT-4o-mini Finishes 0-shot False	"The event 'Unpacking Europe' occurred between year 2001 and year 2002. The event 'Intercontinental Handball Cup' occurred between year 1998 and year 2002. Did 'Unpacking Europe' end in the same year as 'Intercontinental Handball Cup' began? Answer True or False."	"True. The event 'Unpacking Europe' ended in 2002, and the 'Intercontinental Handball Cup' began in 1998. Therefore, 'Unpacking Europe' ended in the same year that 'Intercontinental Handball Cup' began."
2	Phi-3-mini Finished-By 0-shot True	"The event 'Estonian War of Independence' occurred between year 1918 and year 1920. The event 'Syrian National Congress' occurred between year 1919 and year 1920. Did 'Estonian War of Independence' begin before 'Syrian National Congress' began and end in the same year as 'Syrian National Congress'? Answer True or False."	"False. The Estonian War of Independence began in 1918 and ended in 1920, while the Syrian National Congress took place in 1919. Although both events ended in the same year, the Estonian War of Independence began before the Syrian National Congress."
3	Llama-3.1-8B Before CoT False	"The event 'War of the Generals' occurred between year 1911 and year 1912. The event 'Black Power Revolution' occurred between year 1968 and year 1970. Did 'War of the Generals' occur after 'Black Power Revolution' without any overlap between the two events? Answer True or False."	"To determine if 'War of the Generals' occurred after 'Black Power Revolution' without any overlap, let's analyze the information step by step: 1. 'Black Power Revolution' occurred between 1968 and 1970. 2. 'War of the Generals' occurred between 1911 and 1912. 3. Since 'Black Power Revolution' occurred between 1968 and 1970, it started after 1911 and ended before 1912. 4. Therefore, 'War of the Generals' occurred after 'Black Power Revolution' and there was no overlap between the two events. The answer is True."
4	Mistral-7B End-Timepoint 0-shot True	"'Event A' started in 1948. 'Event A' took exactly 39 years. Did 'Event A' end in the year 1987? Answer True or False."	"False. If 'Event A' started in 1948 and took exactly 39 years, then it would have ended in the year 1987 plus 39 years, which is 2026."
5	Mistral-7B Next-occurence 1-shot False	"'Event A' first occurred in year 1773. 'Event A' occurs every 5 years. Did 'Event A' occur again in the year 1779? Answer True or False."	"Answer:True. If 'Event A' occurs every 5 years, then it would have occurred again in the year 1779 (which is 5 years after 1773)."
6	Mixtral-8x7B Next-occurence CoT True	"'Event A' first occurred in year 1555. 'Event A' occurs every 6 years. Did 'Event A' occur again in the year 1561? Answer True or False."	"To determine if 'Event A' occurred again in the year 1561, we need to see if 1561 is a multiple of 6, as we know 'Event A' happens every 6 years. 1. Multiply 6 by any whole number to get 1561: 6 * x = 1561 To find the value of x, divide 1561 by 6: 1561 / 6 $\sim$ 260.167 Since x is a whole number (6 times 256 equals 1536), 'Event A' does not occur in the year 1561. The answer is False."

Table 12: Qualitative examples for failure cases.

Human Alignment: How Much Do We Adapt to LLMs?

Tanguy Cazalets*, **Ruben Janssens***, **Tony Belpaeme, Joni Dambre**
IDLab-AIRO, Ghent University - imec, Ghent, Belgium
{tanguy.cazalets,ruben.janssens,tony.belpaeme,joni.dambre}@ugent.be

Abstract

Large Language Models (LLMs) are becoming a common part of our lives, yet few studies have examined how they influence our behavior. Using a cooperative language game in which players aim to converge on a shared word, we investigate how people adapt their communication strategies when paired with either an LLM or another human. Our study demonstrates that LLMs exert a measurable influence on human communication strategies and that humans notice and adapt to these differences irrespective of whether they are aware they are interacting with an LLM. These findings highlight the reciprocal influence of human-AI dialogue and raise important questions about the long-term implications of embedding LLMs in everyday communication.

1 Introduction and Related Work

Large Language Models (LLMs) enable AI systems to approximate human-like dialogue, significantly expanding the possibilities for human-computer interaction. Their capabilities have become integral to modern life, supporting applications such as educational platforms (Kasneci et al., 2023), physician assistants (Thirunavukarasu et al., 2023), mental well-being support (Ma et al., 2024), and generally extremely personalized user interfaces (Chen et al., 2024). More and more, they are becoming a pervasive presence in our personal worlds, which we interact with in a social manner. However, we don't yet know much about how we adapt to them in these social interactions.

Although many studies focus on how to adapt these models to human needs—through fine-tuning, bias mitigation, or personalization (Navigli et al., 2023; Gallegos et al., 2024; Shum et al., 2018; Ouyang et al., 2022)—fewer have examined how humans adjust their own behavior when interacting with AI (Shen et al., 2024; Woodruff et al., 2024; Floridi and Chiriatti, 2020).

We do know that humans continuously adapt to their conversation partners when communicating. After all, human communication is not simply a passive exchange of information; rather, it is a highly adaptive process (Clark and Brennan, 1991; Ghaleb et al., 2024). In human-human interactions, speakers often engage in *interactive alignment* or *grounding*, converging on vocabulary, syntax, and discourse strategies to optimize clarity and efficiency (Pickering and Garrod, 2013). These adaptations reduce cognitive load and help establish common ground (Clark and Brennan, 1991), thus improving the effectiveness of interpersonal communication. Recent work in cognitive neuroscience even indicates that electrical oscillations in human brains synchronize during meaningful social interactions (Lindenberger et al., 2009; Valencia and Froese, 2020).

It follows logically that humans also adapt to LLMs when interacting with them. If we find that humans consistently shift their language patterns to accommodate AI, this shift may have far-reaching implications for cognition, creativity, and social norms, as previously noted in human-human alignment research (Pickering and Garrod, 2004). Exploring this relationship necessitates a broader interdisciplinary approach, drawing insights from psychology, linguistics, cognitive science, and ethics.

Some research has already investigated how humans adapt to LLMs. This research has largely centered on higher-level cognitive processes such as idea generation (Petridis et al., 2023), scientific writing (Shen et al., 2023), and ethical reasoning (McDonald and Pan, 2020). However, it remains unclear how individuals adapt the lower levels of their cognition to LLMs, such as the language they use and their behaviour in social interactions.

Methodologically, capturing and quantifying mutual adaptation in verbal interaction poses unique

*Equal contribution

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 603–613
July 27 - August 1, 2025

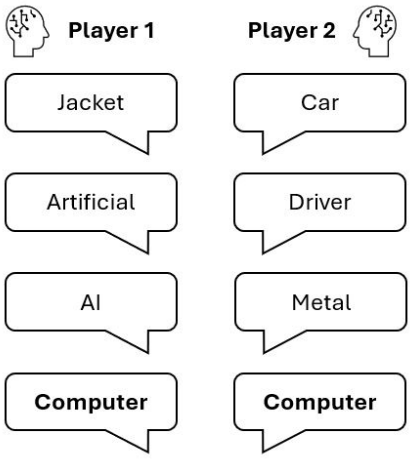


Figure 1: Example of the Word Synchronization Challenge, where participants converge on the same word by the fourth turn.

challenges. While only few studies have looked at how humans adapt to AI systems, alignment in human-human interactions has been a long-standing topic of interest to researchers. However, experimental research in this field has usually studied how humans align their language in reference to some visual information (Garrod and Doherty, 1994; Branigan et al., 2000; Ivanova et al., 2020), while not all human-LLM interactions have visual context. The studies are also limited to lexical and syntactic alignment: they do not study the dynamics of the social interaction.

Yet, social interactions exist of much more than the words that are spoken. Through their choice of words, interlocutors in dialogue share control over the flow of the dialogue. Central to this process is the ability to simulate and predict the other's utterances—part of social cognition (Gandolfi et al., 2022). Do humans use their social cognitive abilities to share control of dialogue with an LLM? And when they do, do they change their behavior compared to when interacting with another human?

1.1 Contribution

In order to study how humans adapt their social behavior and language use to LLMs, we employ a simple language game: the Word Synchronization Challenge (WSC). This game, illustrated in Figure 1, is a multi-turn task where each of two participants (human or artificial) writes down a word, revealed simultaneously at the end of each turn. They aim to converge on the same word as quickly as possible, while not being allowed to use any word previously used by either participant. The game was recently introduced by Cazalets and Dambre (2025), who used it to study LLM-LLM adaptation, but it is also known as an improvisational theater exercise called Convergence or Mind Meld (Hall, 2014), similar to prior work in natural language processing and cross-cultural inference that was inspired by the game Codenames (Kim et al., 2019; Shaikh et al., 2023).

This game constitutes an extremely simple social interaction, not relying on any other modality than verbal interaction, but it requires the two players to coordinate by simulating each other's word associations and aligning their word choices. Convergence in fewer turns is indicative of stronger mutual alignment. The word choices themselves can also be studied: through analyzing the similarity of the chosen words, and the relationships between them, we can study how both players adapt to each other. Furthermore, we study whether any difference in alignment behavior is due to the behavior of the LLM, or because the human is aware they are communicating with an AI model.

In this paper, we address the need to quantify human adaptation to LLMs through the following contributions: (1) introducing the Word Synchronisation Challenge as experimental paradigm to study human-LLM adaptation, (2) studying the extent to which humans align word choices differently with LLMs than they do with humans, (3) studying to which extent this difference is due to the human's awareness of the artificial nature of the LLM, and (4) discussing the potential ethical ramifications for designing AI systems that preserve the richness of human language and cognition.

2 Methods

2.1 Experimental Design

We set up a study where human participants played the Word Synchronization Challenge with both other human players and an LLM. The study used a within-subjects 2x2 factorial design, where we manipulated two factors: whether the participant played against a human or an LLM, and whether the partner was shown to be a human or LLM. This yields the following four conditions:

1. **vs-Human (Human shown):** partner was shown as a human and was indeed a human.

2. **vs-Human (AI shown):** partner was shown as an AI but was in fact a human.

3. **vs-LLM (AI shown):** partner was shown as an AI and was indeed an LLM.

4. **vs-LLM (Human shown):** partner was shown as a human but was in fact an LLM.

Participants completed 4 games per condition (16 games total), with the order of conditions randomized, enabling us to disentangle the effects of actual versus perceived partner identity.

2.2 LLM Implementation

We used OpenAI's GPT-4o model to generate the AI partners responses. The prompt was designed to ensure that the LLMs responses felt natural in the context of the game. In the first round, the prompt encouraged a creative yet random word choice, while subsequent rounds used a dynamic prompt that referenced previous words. Detailed information about the prompt templates, including the rationale behind the design of the prompts, and model settings, is provided in Appendix B.

2.3 Participants

Participants were recruited via Prolific, ensuring a diverse sample of L1 English speakers located in the United Kingdom. A total of 20 participants (6 identified as male, 12 female, and 2 who preferred not to disclose gender; mean age = 34.2 years, SD = 13.05) were enrolled. Participants were compensated GBP 6.90 for their participation, with a median completion time of 48 minutes and 1 second. This payment was considered adequate based on the prevailing market rates in the United Kingdom.

2.4 Procedure

Participants were informed when starting the study that they would complete 16 games with a human or AI player in random order (see instructions in Appendix C). In each game, both players initially entered a random word. In subsequent rounds, they submitted a new, unused word simultaneously, with the game concluding once both players entered the same word, or after a maximum of 16 rounds.

2.5 Post-Game Questionnaire

After each game, participants completed a short questionnaire assessing their experience and strategy use. They rated their partners performance, perceived strategy, and mutual understanding on a 5-point scale (1 representing the lowest performance and 5 the highest). Additionally, they reported their sense of connection with their partner. These self-reported measures complemented our behavioral and linguistic data.

2.6 Ethical Considerations and Data Handling

The study was conducted in accordance with the General Ethical Protocol for research with human participants of our institution, and all data were stored securely. Data were anonymized by assigning each participant a unique randomly generated `playerId`. No personally identifiable information (e.g., IP addresses) was collected.

3 Results and Analysis

3.1 Dataset Filtering and Cleanup

We filtered the data to remove incomplete sessions and other anomalies (e.g., games completed in 2 or fewer rounds, as these were indicative of users repeating a previously used word pattern). The resulting dataset comprised 89 valid H-vs-H games and 139 valid H-vs-LLM games (see Table 1 for details).

3.2 Convergence Metrics

Condition	N	Avg. rounds	Win Rate
vs-LLM (AI shown)	72	8.5	75%
vs-LLM (H shown)	67	8.3	67%
vs-LLM (all)	139	8.4	72%
vs-H (AI shown)	39	6.0	79%
vs-H (H shown)	50	6.8	76%
vs-H (all)	89	6.4	78%

Table 1: Summary of valid games analyzed. We abbreviate Human as H and Artificial Intelligence as AI.

A first high-level indicator is how often the participants successfully converged within 16 rounds and, if they did, how many rounds they needed. Table 1 displays both metrics. A χ^2 test did not reveal any significant differences between the success rates of the four conditions ($p = .63$) or between all human-human and human-LLM games ($p = .35$).

However, when comparing the convergence time for successful games, a Mann-Whitney U-test shows a significant difference ($p < 0.01$) between all human-LLM and human-human games. Within these games, the Mann-Whitney U-tests did not show statistical differences between whether the LLM ($p = 0.64$) or the human partner ($p = 0.27$) were portrayed as AI or human.

3.3 Strategy Analysis

To quantify the convergence strategies used by the participants, we used a linguistic analysis of the

relationships between the words used, and a subjective assessment by the participants themselves. We also qualitatively discuss the trajectory found in one example game in detail, to illustrate the convergence strategies.

3.3.1 Conceptual Linking Score

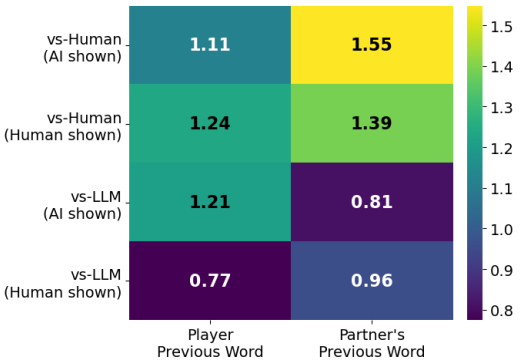


Figure 2: Average CL scores. Each cell represents the average score from word to previous word within a given game configuration.

We computed a Conceptual Linking (CL) score by querying the ConceptNet API (Speer et al., 2017). This score is intended to capture, each round, how semantically related a players current word is to either their own previous word or to their partners previous word. ConceptNet provides weighted associations, and the CL score corresponds to the highest weight found (0 if none). A higher score indicates stronger thematic or conceptual continuity between word choices.

For each game, we computed the average CL scores to both the player's and the partner's word from the previous round, over all rounds. Those averages were then averaged across all games within each configuration, and presented in Figure 2.

While the Mann-Whitney U-test did not reveal any statistical differences between the human-human and human-LLM games for CL score with the player's own previous word ($p = 0.27$), it did show a significant difference when looking at the partner's previous word ($p < 0.001$). No significant differences were found between whether the partner was portrayed as AI or human.

3.3.2 User-perceived Partner Strategy

Following the framework of Cazalets and Dambre (2025), a post-game questionnaire asked participants about their perception of their partners strategies, asking them to choose between: *mirroring* (choosing a word close to the partner's previous word), *staying close* (choosing a word close to their own previous word), or *averaging* (choosing a word halfway between the two previous words).

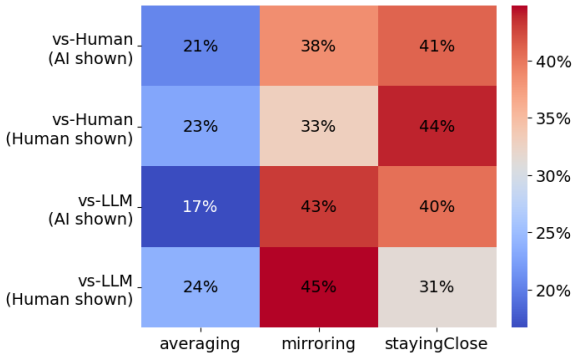


Figure 3: Average reported strategy measures by game configuration. Each cell shows the percentage of time a given strategy was attributed to other player for each game configuration.

Figure 3 shows the average aggregated results of the strategies as reported by the players. A χ^2 test between the four conditions did not reveal a significant difference in user-perceived strategies ($p = 0.56$), and neither when comparing human-human interactions with human-LLM interactions, regardless of how the partner is presented to the player ($p = 0.33$).

3.3.3 Qualitative Illustration of Convergence Trajectory

Figure 4 presents both the word-by-word interaction between a human player and an LLM during a game and the corresponding trajectory of their exchanges in semantic space, with embeddings calculated by word2vec (Mikolov et al., 2013).

This specific game illustrates how both agents adapt their choices based on each other's previous moves. For example, when the player shifts from "sunshine" to "stairs" and "step", the LLM responds with semantically related locations like "rays", and "basement", gradually bridging concepts associated with light, darkness, and structure. In round 3, the LLM's choice of "basement" and the player's move to "dark" signals an attempt to align on the theme of underground, less illuminated spaces. In this example, both partners ultimately settle on the shared concept of "door," indicating a point of semantic agreement.

By projecting the embedding trajectories of these exchanges into three dimensions using PCA (shown in the bottom three views of the Figure 4), we observe how the players word selections nav-

Player	sunshine ☀️	stairs 🪜	step 🪜	dark ⚫	loft 🏠	hatch 🏠	door 🏠
LLM	cellar 🏠	rays ☀️	basement 🏠 ⚫	ladder 🪜 🏠	shadow ⚫	attic 🏠	door 🏠

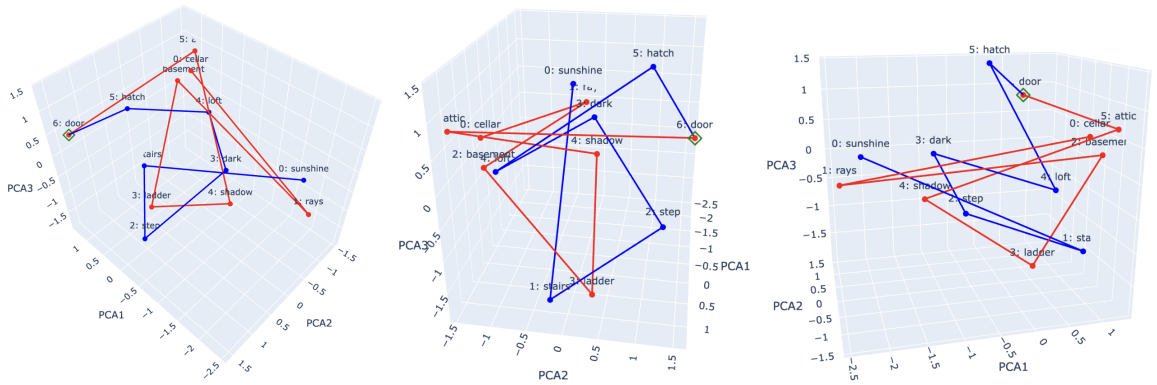


Figure 4: (Top) Table showing the sequence of words exchanged during a game between a Player and an LLM, color-coded by semantic group. (Bottom) Three different views of the projection of the embedding of one game between a human (blue) and a LLM (red). The final word is highlighted with a diamond shape.

igate through semantic clusters—such as moving from themes of light (sunshine, rays) and elevation (stairs, ladder) towards enclosed or connected spaces (loft, attic, door). The visualization highlights periods of alignment, thematic switching, and convergence attempts.

This interaction shows that LLMs adapt and attempt to synchronize with human players over successive rounds, as the human player does with the LLM, and illustrates the strategic, adaptive behaviors emerging from the WSC.

4 Discussion and Conclusion

This study investigated how humans adapt their behavior in social interactions to LLMs. We employed the Word Synchronisation Challenge (WSC) as a "minimal", very simple verbal social interaction, which requires two players to align their word choices, employing social cognition capabilities such as simulating the other's word associations and sharing control of the interaction.

Our results provide evidence that humans do change their behavior in a social interaction with an LLM, compared to how they behave when interacting with another person. Players converged in significantly fewer turns when playing with a human than when playing with an LLM. Furthermore, we analyzed which strategies the players employ in order to converge. Quantifying the semantic similarity between the chosen words through CL scores, we saw that humans changed how they chose their words depending on their partner: when playing against an LLM, they chose words that were significantly less similar to their partner's previous word, compared to when playing against another human. This shows that the difference in convergence rate is linked with a difference in alignment behavior. This difference could be explained by players noticing that the LLM behaves differently—moving more towards their word than a human would—and reacting to this by choosing to stay close to their own word and letting the LLM converge to them. Interestingly, none of these metrics differed when their partner was portrayed as AI or human, indicating the change in adaptation behavior is a reaction to the LLM's behavior rather than to the perception that it is artificial.

In conclusion, our study shows that humans adapt differently to LLMs than to humans in at least some interactions, and that human adaptation happens irrespective of whether they are aware they are interacting with an LLM. With these results in a "minimal social interaction", we make a case for future, deeper research investigating the dynamics of how humans adapt to LLMs, which we believe is an under-researched area of significant importance. As AI systems become increasingly integrated into daily communication, understanding these bidirectional effects is crucial for designing technologies that enhance, rather than constrain, the diversity of human communication. More research is needed, and should focus on the long-term cognitive, social, and cultural implications of these shifts, informing both technological innovation and policy decisions aimed at fostering a balanced co-evolution of human and artificial communicative practices.

Acknowledgments

This project has received funding from the European Unions Horizon 2020 research and innovation program under the Marie Skodowska-Curie grant agreement No 860949, the Flemish Government (AI Research Program), and the Horizon Europe VALAWAI project (grant agreement number 101070930).

Limitations

Our study faces several limitations that warrant consideration. First, the sample size is relatively small and restricted to a narrow demographic, potentially limiting the generalizability of our findings. Variations in individual linguistic proficiency and cultural background may still introduce confounds, suggesting that larger samples are necessary to validate the robustness of these effects.

One notable limitation of our study is inherent to the specificity of the Word Synchronization Challenge. This highly controlled task is both a bug and a feature: while its constrained nature may limit the generalizability of our findings to more spontaneous or naturalistic settings, it also enables precise quantification of alignment effects that might otherwise be obscured in less structured interactions. Our experimental design—though rigorously controlled—cannot fully capture the wide range of spontaneous, real-world conditions under which human-AI dialogue occurs. In particular, the short duration of the Word Synchronization Challenge may not reflect the complexities of natural conversation or the long-term evolution of shared linguistic habits, which could influence both the emergence and persistence of alignment phenomena over time.

It should be noted that LLMs are known to have vocabularies and frequency profiles that diverge from those of human speakers (Yakura et al., 2024). Part of the slower convergence observed with LLM partners may simply reflect this underlying distributional mismatch, rather than the social dynamics we aim to study. However, such differences in vocabulary distributions are inherent to any pair of speakers. Even among humans, active vocabularies can differ substantially, and the aim of the WSC is explicitly to encourage interlocutors to bridge such lexical gaps and to study how they respond to this mismatch by adapting to each other. Our quantitative and qualitative analysis of the adaptation strategies show there is evidence of at least some interactive alignment, rather than the slower convergence only being a result of different vocabulary distributions. Yet, future research should further investigate the impact of these vocabulary distributions, inherent to the chosen model or to the prompt that is used, and strive to better isolate these effects—potentially by controlling or measuring the LLMs distribution more explicitly or comparing with humans with more or less different vocabularies.

Finally, our metrics for quantifying convergence may overlook nuanced pragmatic or syntactic adaptations. Future studies could expand these methods to incorporate richer dialogue annotation, or longitudinal tracking of individual language changes to provide a more comprehensive view of human-LLM co-adaptation.

Ethical Considerations

The divergence seen in human-LLM pairs raises questions about long-term implications of embedding LLMs in daily life. At a societal level, the homogenization of language and thought is a valid concern, particularly if users unconsciously pick up machine-like expressions or patterns. While when considering human-human alignment, some degree of efficiency can be beneficial, a loss of linguistic diversity may undercut creativity and cultural specificity. This underscores the importance of AI literacy initiatives that educate users about potential shifts in their communicative styles when relying heavily on AI systems.

Resources

The codebase and data for reproducing our experiments can be accessed at: `https://github.com/Finebouche/words_synch_challenge`

A demo is available on the project website: `https://word-sync.games/`

Author Contributions

Conceptualization, implementation, data collection: T.C.; **Experiment design and statistical analysis:** R.J.; **Data analysis and manuscript writing:** T.C. and R.J.; **Supervision and manuscript review:** T.B. and J.D.

All authors have read and approved the final version.

References

Holly P Branigan, Martin J Pickering, and Alexandra A Cleland. 2000. Syntactic co-ordination in dialogue. *Cognition*, 75(2):B13–B25.

Tanguy Cazalets and Joni Dambre. 2025. Word synchronization challenge: A benchmark for word association responses for llms. *Preprint*, arXiv:2502.08312.

Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, et al. 2024. When large language models meet personalization: Perspectives of challenges and opportunities. *World Wide Web*, 27(4):42.

Herbert H. Clark and Susan E. Brennan. 1991. *Grounding in communication.*, pages 127–149. American Psychological Association.

Luciano Floridi and Massimo Chiriatti. 2020. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, pages 1–79.

Greta Gandolfi, Martin J Pickering, and Simon Garrod. 2022. Mechanisms of alignment: shared control, social cognition and metacognition. *Philosophical Transactions of the Royal Society B*, 378(1870):20210362.

Simon Garrod and Gwyneth Doherty. 1994. Conversation, co-ordination and convention: an empirical investigation of how groups establish linguistic conventions. *Cognition*, 53(3):181–215.

Esam Ghaleb, Marlou Rasenberg, Wim Pouw, Ivan Toni, Judith Holler, Asl Özyürek, and Raquel Fernández. 2024. Analysing cross-speaker convergence in face-to-face dialogue through the lens of automatically detected shared linguistic constructions.

William Hall. 2014. *The Playbook: Improv Games for Performance*.

Iva Ivanova, William S Horton, Benjamin Swets, Daniel Kleinman, and Victor S Ferreira. 2020. Structural alignment in dialogue and monologue (and what attention may have to do with it). *Journal of Memory and Language*, 110:104052.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274.

Andrew Kim, Maxim Ruzmaykin, Aaron Truong, and Adam Summerville. 2019. Cooperation and codenames: Understanding natural language processing via codenames. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 15, pages 160–166.

Ulman Lindenberger, Shu-Chen Li, Walter Gruber, and Viktor Müller. 2009. Brains swinging in concert: cortical phase synchronization while playing guitar. *BMC neuroscience*, 10:1–12.

Zilin Ma, Yiyang Mei, and Zhaoyuan Su. 2024. Understanding the benefits and challenges of using large language model-based conversational agents for mental well-being support. In *AMIA Annual Symposium Proceedings*, volume 2023, page 1105.

Nora McDonald and Shimei Pan. 2020. Intersectional ai: A study of how information science students think about ethics and their impact. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–19.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Roberto Navigli, Simone Conia, and Björn Ross. 2023. Biases in large language models: origins, inventory, and discussion. *ACM Journal of Data and Information Quality*, 15(2):1–21.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Savvas Petridis, Nicholas Diakopoulos, Kevin Crowston, Mark Hansen, Keren Henderson, Stan Jastrzebski, Jeffrey V Nickerson, and Lydia B Chilton. 2023. Anglekindling: Supporting journalistic angle ideation with large language models. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pages 1–16.

Martin J. Pickering and Simon Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27(02).

Martin J. Pickering and Simon Garrod. 2013. An integrated theory of language production and comprehension. *Behavioral and Brain Sciences*, 36(4):329–347.

Omar Shaikh, Caleb Ziems, William Held, Aryan Pariani, Fred Morstatter, and Diyi Yang. 2023. Modeling cross-cultural pragmatic inference with codenames duet. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6550–6569.

Hua Shen, Chieh-Yang Huang, Tongshuang Wu, and Ting-Hao Kenneth Huang. 2023. Convxai: Delivering heterogeneous ai explanations via conversations to support human-ai scientific writing. In *Companion Publication of the 2023 Conference on Computer Supported Cooperative Work and Social Computing*, pages 384–387.

Hua Shen, Tiffany Knearem, Reshmi Ghosh, Kenan Alkiek, Kundan Krishna, Yachuan Liu, Ziqiao Ma, Savvas Petridis, Yi-Hao Peng, Li Qiwei, Sushrita Rakshit, Chenglei Si, Yutong Xie, Jeffrey P. Bigham, Frank Bentley, Joyce Chai, Zachary Lipton, Qiaozhu Mei, Rada Mihalcea, Michael Terry, Diyi Yang, Meredith Ringel Morris, Paul Resnick, and David Jurgens. 2024. Towards bidirectional human-ai alignment: A systematic review for clarifications, framework, and future directions. *Preprint*, arXiv:2406.09264.

Heung-Yeung Shum, Xiaodong He, and Di Li. 2018. From eliza to xiaoice: Challenges and opportunities with social chatbots.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature medicine*, 29(8):1930–1940.

Ana Lucía Valencia and Tom Froese. 2020. What binds us? inter-brain neural synchronization and its implications for theories of human consciousness. *Neuroscience of Consciousness*, 2020(1):niaa010.

Allison Woodruff, Renee Shelby, Patrick Gage Kelley, Steven Rousso-Schindler, Jamila Smith-Loud, and Lauren Wilcox. 2024. How knowledge workers think generative ai will (not) transform their industries. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI 24, pages 1–26. ACM.

Hiromu Yakura, Ezequiel Lopez-Lopez, Levin Brinkmann, Ignacio Serna, Prateek Gupta, and Iyad Rahwan. 2024. Empirical evidence of large language model's influence on human spoken communication. *arXiv preprint arXiv:2409.01754*.

A Data Collection through a Web app

A.1 Overview

In this study, participants used a custom Web application (developed in JavaScript and Node.js) to play the "Word Synchronization Challenge." (Cazalets and Dambre, 2025). Figure 6 shows a screenshot of the game interface during play.

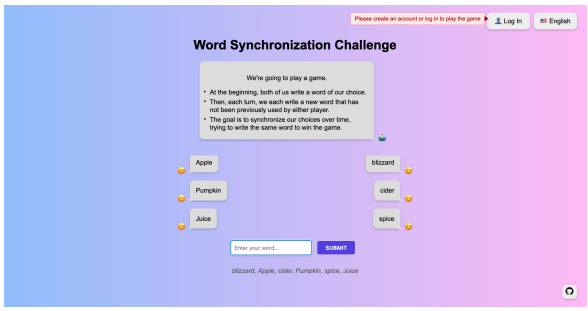


Figure 5: Screenshot of the web app during a game with another human

A.2 Application Architecture

The back end consists of a Node.js/Express server that handles HTTP requests, user sessions, and game-related APIs; AaSocket.io module that enables real-time communication and game state synchronization for human-human games: a SQLite database managed with Sequelize, which stores persistent game and user data.

Our database schema defines two models:

- **Player:** Stores each users `playerId` and prolific Id.
- **Game:** Records game details, including the `playerIds` involved (or a `botId`, when playing with an LLM), language settings, sequence of played words, number of rounds, the winning player, and post-game survey responses.

B LLM Model Prompt Engineering

To standardize interactions with the LLM during the word game, we designed specific prompts for each round. Our approach encourages diversity in the first response and guided, context-aware adaptation in subsequent rounds. Below, we detail the prompts used:

B.1 Initial Round (Round 1)

```
Round 1. New game, please give
your first (really random) word
and only that word. You can be a
bit creative but not too much. Be
sure to finish your answer with
it.
```

Rationale: This prompt instructs the LLM to provide a single, moderately creative word as its initial response, setting a baseline for the game without strong contextual bias to avoid predictable or semantically anchored openings.

B.2 Subsequent Rounds (Rounds 2 and onward)

```
"${player_word}! We said
different words, let's do
another round. So far
we have used the words:
[${past_words_array.join(',
')}], they are now forbidden.
Based on previous words, what
word would be most likely for
next round given that my word
was ${player_word} and your word
was ${bot_word}? Please give
only your word for this round."
```

Rationale: For each subsequent round, the prompt dynamically incorporates both players prior words (explicitly listing them as forbidden to not break the rules) and provides the most recent player and bot words as context. The LLM is explicitly asked to generate the next word based on this shared context, fostering both semantic continuity and adaptation. The instruction to output only your word ensures concise, focused responses.

B.3 Model Settings

We adjusted the model settings based on the round number to ensure varied yet contextually constrained responses. The settings are as follows:

Round	Temperature	Max Tokens
Round 1	1.6	50
Other Rounds	1.1	20

Table 2: Model settings for different rounds.

These settings and the prompt design were critical in ensuring that the models responses were both natural and aligned with the games requirements. Detailed implementation code is available upon request.

C Instruction given to participants

In this study, you will participate in a simple word-guessing cooperative game

At the beginning, each player writes a random word. Then, on each turn, each player writes a new word that has not been previously written. The goal is to produce the same word, in which case the game is won. If the game exceeds 15 turns, the game is lost.

Connection (1 time)

1. When you arrive on the website, please click on **"Log in" "Generate ID"**.
2. Copy this ID and **"Log in"** using it.
3. Fill in the requested information carefully (some are redundant with Prolific).
4. **Important:** Ensure you fill in your Prolific ID.

Play the game (16 times)

1. You will play **8 games with an Artificial Intelligence (AI)** system and **8 games with another human player**, in random order.
2. Press the button **"Play with a human"** or **"Play with an AI"** to start the game.
3. It might take some time before the AI system is ready or before another human player joins. If you have to wait more than 3 minutes, refresh the page and click the **"Play with..."** button again.
4. After each game, you will be asked to fill in a questionnaire (see below).

Fill in the questionnaire (16 times) After each game, youll be asked a few questions about your experience. These questions help us understand how you and your partner strategized during the game. Specifically, youll be asked to share:

- **Your Strategies:** What approach or idea you used while playing.
- **Your Partners Strategies:** What you think your partner was trying to do.
- **Whether you believe your partner understood the strategy you used.**
- **Whether you understood your partners strategy.**
- **Partner Rating:** Rate your partners performance on a scale from 1 (lowest) to 5 (highest).

Your answers will help us learn more about how interacting with AI versus human partners influences communication and decision-making.

Upon completion of all the games and questionnaires, you will be given a link to complete the study in Prolific. Thanks for playing!

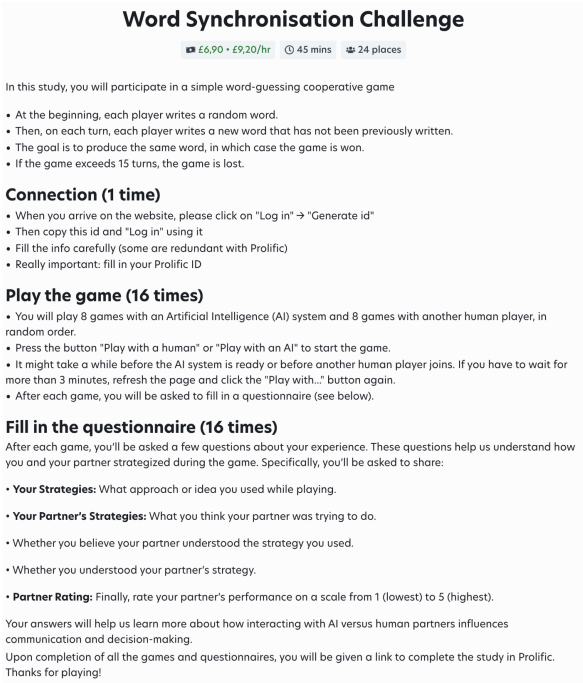


Figure 6: Screenshot of the instruction as seen in prolific

D Responsible use of AI

GitHub Copilot and OpenAI ChatGPT have been used as coding assistants for website implementation and data analysis. All code generated by AI assistants was manually reviewed.

OpenAI ChatGPT was used to correct grammar and typos in writing. All AI-generated text was manually reviewed.

E Licenses

This work includes data from ConceptNet 5, which was compiled by the Commonsense Computing Initiative. ConceptNet 5 is freely available under the Creative Commons Attribution-ShareAlike license (CC BY SA 4.0) from https://conceptnet.io. The included data was created by contributors to

Commonsense Computing projects, contributors to Wikimedia projects, Games with a Purpose, Princeton University's WordNet, DBPedia, OpenCyc, and Umbel.

The Word Synchronization Challenge framework is licensed under the MIT license, a permissive open-source license that allows for unrestricted reuse, modification, and distribution, including for commercial purposes, provided that the original copyright notice and license are included.

Dynamic Order Template Prediction for Generative Aspect-Based Sentiment Analysis

Yonghyun Jun and **Hwanhee Lee***
Department of Artificial Intelligence, Chung-Ang University
{zgold5670, hwanheelee}@cau.ac.kr

Abstract

Aspect-based sentiment analysis (ABSA) assesses sentiments towards specific aspects within texts, resulting in detailed sentiment tuples. Previous ABSA models often used static templates to predict all the elements in the tuples, and these models often failed to accurately capture dependencies between elements. Multi-view prompting method improves the performance of ABSA by predicting tuples with various templates and then assembling the results. However, this method suffers from inefficiencies and out-of-distribution errors. In this paper, we propose a Dynamic Order Template (DOT) method for ABSA, which dynamically creates an order template that contains only the necessary views for each instance. Ensuring the diverse and relevant view generation, our proposed method improves F1 scores on ASQP and ACOS datasets while significantly reducing inference time.*

1 Introduction

Aspect-based sentiment analysis (ABSA) aims to identify the sentiment of aspects in a given text rather than simply classifying the overall sentiment of the entire text. ABSA research evolves to generate quadruples consisting of four elements: 1) Aspect (A), 2) Category (C) for the type of A, 3) Opinion (O) for A, and 4) Sentiment (S) for A. Many recent studies, such as T5-paraphrase, tackle this problem using generative models (Zhang et al., 2021b). These approaches usually get review sentences as input and output the span of quadruples in fixed order form, such as "C is S because A is O" (Zhang et al., 2021a). However, this static single-order template cannot express the dependence between elements as in Figure 1 due to the autoregressive nature of the transformer (Vaswani et al., 2017). Moreover, the model output can heavily depend on the order of generation of each element (Hu et al., 2022).

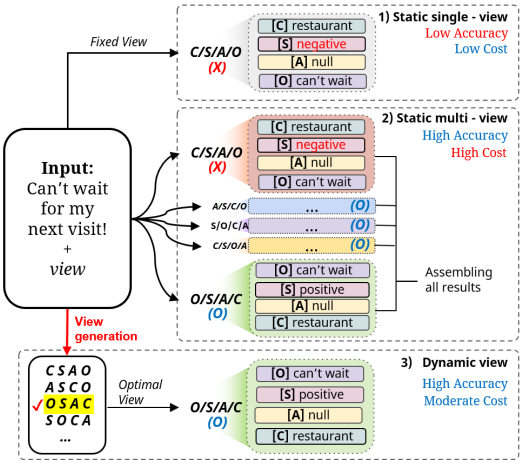


Figure 1: Comparison of three different generative ABSA methods. 1) static single-view, 2) static multi-view, and 3) dynamic-view prediction (ours).

Multi-view prompting (Gou et al., 2023) (MvP) deals with this issue by constructing order templates as a channel for "viewing" different perspectives in a sentence. As shown in Figure 1, MvP permutes all possible element orders and sorts them based on the entropy of the pre-trained model at the dataset level. Using this entropy, MvP samples top-k orders and adds these orders as a prompt template. During inference time, MvP conducts majority votes on generated sentiment tuples with various templates. Through this ensemble approach, MvP uses the intuition of solving problems from multiple views in human reasoning and decision (Stanovich and West, 2000), resulting in enhanced performance. However, we find that this static multi-view approach of MvP has several drawbacks: ***1) Inefficiency:*** Even for samples where the answer can be easily found and multiple views are not necessary, this method generates the same number of views, resulting in unnecessary computation that increases the inference

*Corresponding author.
*Our implementation is publicly available at https://github.com/imsongpasimin/DOT

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 614–626
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

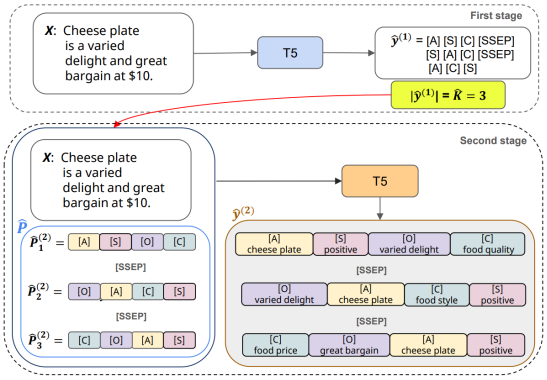


Figure 2: Overview of our proposed two stage method. We use two T5 models for each stage: one for generating the initial order template, the other for forming the final order template and generating sentiment tuples.

time. **2)** ***Limited Transferability***: MvP uses the number of views k as a hyperparameter, applying the same k value across all datasets during training and inference. However, since the optimal number of ensemble models varies according to the data domain, it requires manual adjustment of the k value for each dataset (Shahhosseini et al., 2022), which hinders the transferability to other datasets. To resolve the aforementioned shortcomings, we propose a Dynamic Order Template (DOT) method for ABSA that combines the advantages of both single-view and multi-view approaches. By prioritizing multiple views based on instance-level entropy, DOT aims to generate only the necessary number of views for each instance during inference. For an example that contains only one tuple, as in Figure 1, DOT dynamically creates only one view as an order template necessary to predict the tuple. After generating the views, DOT generates tuples using the views inside the order template. This phase operates in a multi-view manner, enabling us to retain the benefits of previous multi-view methods. Extensive experiments on four widely used sentiment quadruple prediction datasets, derived from ASQP (Pontiki et al., 2016; Zhang et al., 2022), ACOS (Cai et al., 2021a, 2023), and MEMD-ABSA (Cai et al., 2023), demonstrate that our method shows state-of-the-art performance with significantly lower inference time compared to the multi-view approach. Moreover, we show that our method is robust to domain shift compared to previous methods, resulting in higher transferability.

2 Method

Our proposed Dynamic Order Template (DOT) method is composed of two stages as in Figure 2. The first stage involves generating an initial order template (Sec 2.1) to predict the number of tuples. The second stage involves refining the initial template from stage 1 to produce the final order template and predicting the sentiment tuples based on it (Sec 2.2). For both stages, we map sentiment tuples (A, C, S, O) to marker tokens $[A]$, $[C]$, $[S]$, and $[O]$ respectively. Also, for the instances that contain multiple sentiment tuples, we indicate each tuple with the respective tokens and concatenate the targets with [SSEP] tokens. It is important to note that null or missing values in the datasets are encoded as the literal string "null". During template construction, our method maps this string directly to the designated marker tokens, eliminating the need for any special handling.

2.1 Stage 1: Generating Order Template

We assume that the number of sentiment tuples K_i in i^{th} instance present for each instance corresponds to the required number of views. Consequently, we are able to divide the complex sentiment tuple generation task into two relatively simpler subtasks: 1) predicting the number of sentiment tuples in the sentence, and 2) generating sentiment tuples with the exact number of first stage predictions. Building separate models specialized for each subtask and enabling collaboration between them was instrumental in achieving significant performance improvements. In this framework, each view is interpreted as the ordered scaffold for generating a single sentiment tuple, as depicted in Figure 2. This allows each prediction order to correspond one-to-one with a sentiment tuple in the second stage.

We observe that instead of directly using the value of K_i as the target, sampling views corresponding to K_i as the target for the model to generate leads to more accurate prediction of the number of tuples in the given instance (More details are in Appendix C). To establish the view sampling strategy, we start by ranking all possible views generated through permutations through each entropy score, following (Hu et al., 2022). Specifically, we calculate entropy of each view v in instance-level with vanilla T5 by calculating conditional generation probability as follows:

$$\mathcal{E}_{i,v} = -\sum P(v|x_i) \log P(v|x_i) \quad (1)$$

Here, $\mathcal{E}_{i,v}$ is the entropy of the total sequence when the i_{th} instance is input into the T5 model and v

is the output. At this time, we note that actually utilizing only A, C, S during the first stage notably facilitates the training process in the second stage. We provide a detailed analysis on excluding O in Appendix C.2. After computing the entropy, we sort the views by the entropy in ascending order to get the ranked set of views $P_i^{(1)}$. And then we sample the top K_i views for each sample and concatenate these views as an order template. Using the original i^{th} input sentence, we train the T5 model to generate the first-stage target $y_i^{(1)}$ as follows:

$$y_i^{(1)} = P_{i,1}^{(1)} \text{ [SSEP] } P_{i,2}^{(1)} \text{ [SSEP] } \ldots P_{i,K_i}^{(1)},$$

where $P_{i,K_i}^{(1)}$ denotes K_i^{th} view in $P_i^{(1)}$. We set the loss function to train the T5 model as in Equation (2), where $|B|$ denotes the batch size of the model. The scaling factor is omitted for simplicity.

$$\mathcal{L}_1 = -\sum_{i=1}^{|B|}\sum_{t=1}^{T} \log p(\boldsymbol{y_{i,t}^{(1)}}|\boldsymbol{x_i}, \boldsymbol{y_{i,<t}^{(1)}}) \quad (2)$$

2.2 Stage 2: Sentiment Tuple Generation

In the second stage, the model is trained to generate the sentiment tuple of a given instance using the number of sentiment tuples (i.e. K_i). Different from the first stage, we need to generate all elements in sentiment quadruples including O in this stage. Hence, we re-rank all views to pick K_i views including O (i.e. (A, C, S, O)). Here, we adopt the same strategy as in the first stage, using entropy to form a ranked set of views, $P_i^{(2)}$. We then sample top K_i views from $P_i^{(2)}$ and add them as an order template prompt P_i to original input sentence. We design the second stage target $y_i^{(2)}$ by aligning each sentiment tuple with an order template, ensuring that the model learns to generate different tuples for different perspectives. Also, we place the corresponding elements next to each marker token within P_i as follows:

$$y_i^{(2)} = P_{i,1}^{(2)} \otimes tuple_1 \text{ [SSEP] } \ldots P_{i,K_i}^{(2)} \otimes tuple_{K_i},$$

where $P_{i,K_i}^{(2)}$ represents K_i^{th} view in $P_i^{(2)}$ and $tuple_{K_i}$ is the K_i^{th} sentiment tuple for given instance. $\otimes$ denotes an interleaved combination between marker tokens and elements. Detailed examples for both stages are present in Appendix E. We design the loss function for training the T5 model in second stage as follows.

$$\mathcal{L}_2 = -\sum_{i=1}^{|B|}\sum_{t=1}^{T} \log p(\boldsymbol{y_{i,t}^{(2)}}|\boldsymbol{x_i}, \boldsymbol{P_i}, \boldsymbol{y_{i,<t}^{(2)}}) \quad (3)$$

2.3 Two-stage inference

During inference time, two stages are conducted sequentially. In the first stage, the model generates the initial order template, denoted as $\hat{y^{(1)}}$. In the second stage, we count the number of generated views from $\hat{y^{(1)}}$ to set $\hat{K}$. Using $\hat{K}$, we sample the top $\hat{K}$ views from the newly ranked set of views and constructs the final order template, referred to as $\hat{P}$. Finally, $\hat{P}$ is directly appended to the inference sentence, enabling the generation of different sentiment tuples for each view in $\hat{P}$. Throughout inference, we employ a constrained-decoding strategy (De Cao et al., 2020) to ensure that the output at each stage conforms to the required format. The overall two-stage process is described in Figure 2.

3 Experiment

3.1 Benchmark Datasets

We adopt two widely used ABSA datasets: ASQP and ACOS, where the task is to predict sentiment quadruples. For ASQP task, we use rest15 (R15) and rest16 (R16) datasets released from (Pontiki et al., 2016; Zhang et al., 2022). For ACOS task, we use laptop16(Lap) and rest16(Rest) datasets constructed by (Cai et al., 2021a; Pontiki et al., 2016). Also, we adopt additional ACOS benchmarks from MEMD datasets (Restaurant, Laptop, Books, Clothing, Hotel) (Xu et al., 2023) which use a different source from the previous datasets. We refer to the Restaurant and Laptop datasets in MEMD as M-Rest and M-Laptop, respectively, for the sake of clarity.

3.2 Baselines

We benchmark our approach against a suite of extraction- and generation-based baselines. For a subset of baselines, we confine evaluation to the four datasets—R15, R16, Laptop, and Restaurant—because these methods were originally optimized for ASQP or ACOS tasks. Other baselines are assessed on their full applicable dataset range. This selective restriction ensures a fair comparison. The complete list of baselines is as follows: *TAS-BERT* (Wan et al., 2020) jointly extracts and detects sentimental tuples. *Extract-Classify* (Cai et al., 2021b) divide the task into two stages: extraction and classification. *One-ASQP (large)* (Zhou et al., 2023) identify the aspect-opinion-sentiment (AOS) triplets simultaneously. *Seq2Path* (Mao et al., 2022) generates sentiment tuples as multiple paths of a

Methods	ASQP		ACOS		MEMD					Avg	Time(s)
	R15	R16	Lap	Rest	M-Rest	M-Lap	Books	Clothing	Hotel		
TAS-BERT	34.78	43.71	27.31	33.53	-	-	-	-	-	-	-
Extract-Classify	36.42	43.77	35.80	44.61	-	-	-	-	-	-	-
One-ASQP (large)	-	-	41.56	60.69	-	-	-	-	-	-	-
Seq2Path	-	-	42.97	58.41	-	-	-	-	-	-	-
AugABSA	50.01	60.88	-	-	-	-	-	-	-	-	-
SCRAP	49.93	**62.48**	-	-	-	-	-	-	-	-	-
Paraphrase	46.93	57.93	43.51	61.16	57.38	35.07	39.30	43.00	68.79	50.34	40.63
DLO	48.18	59.79	43.64	59.99	57.07	35.56	42.63	43.35	**70.27**	51.16	260.74
MvP	51.04	60.39	43.92	**61.54**	58.12	35.25	42.57	**43.94**	69.06	51.76	2161.81
GPT-4o	40.45	47.29	24.77	46.53	35.11	20.69	30.39	40.27	24.84	34.48	-
LLaMa-3.1-8b	37.52	47.60	40.07	54.06	38.10	31.16	28.62	32.21	44.62	39.33	-
Qwen-2.5-7b	29.93	39.34	12.48	33.56	25.63	24.13	17.77	18.09	38.03	26.66	-
Mistral-7b	44.14	51.96	39.02	53.02	41.28	26.80	26.54	21.81	40.35	38.32	-
DOT (Ours)	**51.91**	61.24	**44.92**	59.25	**58.25**	**39.02**	**43.02**	43.37	69.94	**52.28**	298.17

Table 1: F1 scores for ABSA on nine datasets. Best results are in bold, second-best underlined. Results are averaged over five seeds. Time denotes average inference time.

tree, and automatically selects a valid one. *AugABSA* (Wang et al., 2023) generates a original text based on augmented sentiment quadruples. *SCRAP* (Kim et al., 2024) optimizes the model to generate extract-then-assign reasonings and the corresponding sentiment quadruplets in sequence. *Paraphrase* (Zhang et al., 2021a) formulates a paraphrase generation process for ABSA with a single fixed order. *DLO* (Hu et al., 2022) augments data via the multiple order templates. *MvP* aggregates sentiment tuples generated from multiple orders of prompts via ensembling. Also, we benchmark popular LLMs such as GPT-4o (Hurst et al., 2024), LLaMa-3.1-8b (Dubey et al., 2024), Qwen-2.5-7b (Yang et al., 2025), and Mistral-7b (Jiang et al., 2023). Detailed setups for LLMs are in Appendix H.

3.3 Implementation Details

We utilize the pre-trained T5-base (Raffel et al., 2020) model as the backbone for the first stage. We also use the model trained in the first stage as the backbone for the second stage, allowing us to leverage a tuned initial point for the ABSA dataset to have the regularization effect inspired by (Fu et al., 2023). Additionally, we observe that the label of the datasets (i.e. sentiment tuples) irregularly contains stop words. Thus, we eliminate irregularities in tuples through stop-word filtering in the second stage. We provide a detailed analysis and filtering process in Appendix A.

3.4 Results

Performance Comparison We use the F1 score, which is a standard metric for ABSA, to measure the performance of the systems. Table 1 demonstrates that our DOT framework achieves state-of-the-art results, ranking first on five of the nine benchmarks and second on three others. To further elaborate, a significant factor contributing to the enhanced performance is our approach of dividing the complex sentiment quadruple generation task into two subtasks, each handled by dedicated models. As a result, our method achieves better performance compared to a single model handling multi-view processing alone. Crucially, the effectiveness of this two-stage approach depends on the accuracy of the first task, which we comprehensively presented in Appendix C. However, its performance is slightly lower on the Rest and Clothing datasets, which we analyze in Section F.

Inference time We also measure inference time using T5-base model for all baselines. We check inference time for each dataset, and average them. As in Table 1, we dramatically reduce inference time particularly compared to the multi-view methods such as MvP (Gou et al., 2023), by predicting solely the necessary number of views for each sample. On the other hand, single view inferences (Zhang et al., 2021a; Hu et al., 2022) and extraction approaches (Wan et al., 2020; Cai et al., 2021b; Zhou et al., 2023) can be more memory- and time-efficient(Zhou et al., 2023), they generally exhibit inferior performance compared to generative methods. We provide more details on the inference time in Appendix D.

Transferability To examine the transferability of each model, we conduct an in-depth experiment on cross-dataset evaluation. We group the datasets

Train	*SemEval*		*Yelp*	
Test	*SemEval*	*Yelp*	*Yelp*	*SemEval*
Paraphrase	52.38	38.52 (-11.86)	57.38	44.88 (-12.50)
MvP_3	55.62	34.42 (-21.20)	57.27	41.72 (-15.55)
MvP_9	56.89	35.02 (-21.87)	56.98	42.52 (-14.46)
MvP_{15}	**57.66**	35.21 (-21.45)	58.12	41.94 (-16.18)
DOT	57.47	**39.88** (-17.59)	**58.25**	**46.97** (-11.28)

Table 2: Cross-dataset evaluation results for validating the effect of domain shift.

Model Configuration	Average F1
Full Model	54.33
w/o filtering	53.31 (-1.02)
w/o stage division	52.73 (-1.60)
w/o entropy score	52.53 (-1.80)
w/o multi view	52.31 (-2.02)
w/o stage division, entropy score	50.04 (-4.29)
w/o filtering, stage division, entropy score	45.80 (-8.53)

Table 3: Ablation study for the proposed method, which shows the average F1 across ASQP and ACOS.

into two sources: SemEval (Pontiki et al., 2016) (R15, R16, Rest) and Yelp (M-Rest), and assess performance by training on one group and testing on the other in a zero-shot setting. For the MvP model, we vary the number of views (3, 9, and 15) to evaluate sensitivity in static multi-view methods, while T5-paraphrase uses a static single order. As shown in Table 2, our model significantly outperforms the baselines in cross-dataset evaluation. Although T5-paraphrase suffers a smaller performance drop, it still falls behind our method, and MvP shows notable degradation regardless of view count. These results demonstrate that our model effectively identifies optimal views even for out-of-domain datasets.

Performance Across Model Scales Beyond using T5-base as the sole backbone, we extend our experiments to examine how performance varies with model scale, specifically testing T5-small and T5-large. In this setup, we employ the same backbone model for both stages. As shown in Figure 3, performance consistently improves across all benchmarks as the backbone scale increases, supporting the scaling law (Kaplan et al., 2020) showing larger models yield better accuracy. Nonetheless, these gains do not fully compensate for the inefficiency introduced by loading larger models and performing multiple inference steps. Considering the trade-off between accuracy and computational cost, T5-base emerges as the most balanced choice.

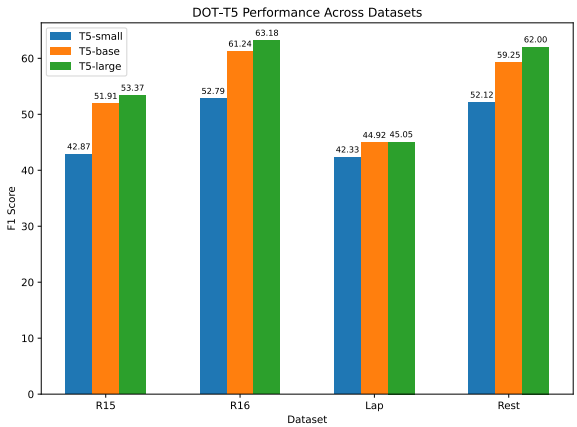


Figure 3: Bar chart illustrating F1 performance variations across T5-small, T5-base, and T5-large backbones on R15, R16, Laptop, and Rest benchmarks.

Ablation study To further investigate the effectiveness of each component of our framework, we conduct an ablation study and present the results in Table 3. Firstly, we record the results without stop-word filtering. Also, we unify the two stages into one, directly generating multiple order templates and tuples without including order prompting. Additionally, we evaluate the results of sampling the views randomly, checking whether the entropy score is valid. Lastly, we exclude the multi-view approach by training and testing our model using only the view with the lowest entropy for each instance as the order template. We perform an ablation study by excluding one or more of the four components of our method mentioned earlier. By observing the gaps between these variants with the original model, we verify the effectiveness of each component of our method. For a more comprehensive ablation study, please refer to Appendix C.

4 Conclusion

We propose Dynamic Order Template (DOT) method for generative ABSA, addressing inefficiencies and out-of-distribution errors. Experiments on nine datasets demonstrate that DOT achieves state-of-the-art performance with reduced inference time, effectively balancing the strengths of previous single and multi-view approaches for ABSA.

Limitation

Our DOT method is highly efficient and powerful, yet it still has several limitations. DOT method consists of two stages: view generation and tuple generation. We train separate models for each task,

and these two models perform inference sequentially. This form is not end-to-end, so it is disadvantageous in terms of training time and memory.

Also, since we directly connect first stage and second stage, if any errors occur, the errors may propagate and magnify as it moves to the subsequent stage. It results in relatively large standard deviation for different seeds as reported in Table 10. However, by splitting the task of 'predicting the appropriate number of tuples' into two sub-tasks—'predicting the appropriate number of tuples' and 'accurately predicting the tuples'—it becomes significantly easier to achieve accurate results in both areas, thereby enhancing overall performance in our work.

Finally, we define the number of necessary views as the number of sentiment tuples for simplicity and efficiency. A more complex yet refined method for determining the necessary number of views could be further explored in future research.

Ethics Statement

This study utilizes the various datasets for aspect-based sentiment analysis, which are accessible online. Additionally, we have properly cited all the papers and sources referenced in our paper. We plan to release the pre-trained model and the code for training the proposed system.

Acknowledgement

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2021-II211341, Artificial Intelligence Graduate School Program (Chung-Ang University)].

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hongjie Cai, Nan Song, Zengzhi Wang, Qiming Xie, Qiankun Zhao, Ke Li, Siwei Wu, Shijie Liu, Jianfei Yu, and Rui Xia. 2023. Memd-absa: a multi-element multi-domain dataset for aspect-based sentiment analysis. *arXiv preprint arXiv:2306.16956*.

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021a. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 340–350.

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021b. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 340–350.

Siva Uday Sampreeth Chebolu, Franck Dernoncourt, Nedim Lipka, and Thamar Solorio. 2023. A review of datasets for aspect-based sentiment analysis. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 611–628.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Alabhya Farkiya, Prashant Saini, Shubham Sinha, and Sharmishta Desai. 2015. Natural language processing using nltk and wordnet. *Int. J. Comput. Sci. Inf. Technol*, 6(6):5465–5469.

Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. 2023. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12799–12807.

Zhibin Gou, Qingyan Guo, and Yujiu Yang. 2023. Mvp: Multi-view prompting improves aspect sentiment tuple prediction. *arXiv preprint arXiv:2305.12627*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Mengting Hu, Yike Wu, Hang Gao, Yinhao Bai, and Shiwan Zhao. 2022. Improving aspect sentiment quad prediction via template-order data augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7889–7900.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Jieyong Kim, Ryang Heo, Yongsik Seo, SeongKu Kang, Jinyoung Yeo, and Dongha Lee. 2024. Self-consistent reasoning-based aspect-sentiment quad prediction with extract-then-assign strategy. *arXiv preprint arXiv:2403.00354*.

M Lewis. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Yue Mao, Yi Shen, Jingchao Yang, Xiaoying Zhu, and Longjun Cai. 2022. Seq2path: Generating sentiment tuples as paths of a tree. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2215–2225.

Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphee De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Mohsen Shahhosseini, Guiping Hu, and Hieu Pham. 2022. Optimizing ensemble weights and hyperparameters of machine learning models for regression problems. *Machine Learning with Applications*, 7:100251.

Marina Solnyshkina, Radif Zamaletdinov, Ludmila Gorodetskaya, and Azat Gabitov. 2017. Evaluating text complexity and flesch-kincaid grade level. *Journal of social studies education research*, 8(3):238–248.

Keith E. Stanovich and Richard F. West. 2000. Advancing the rationality debate. *Behavioral and Brain Sciences*, 23(5):701–717.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Hai Wan, Yufei Yang, Jianfeng Du, Yanan Liu, Kunxun Qi, and Jeff Z Pan. 2020. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9122–9129.

An Wang, Junfeng Jiang, Youmi Ma, Ao Liu, and Naoaki Okazaki. 2023. Generative data augmentation for aspect sentiment quad prediction. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (* SEM 2023)*, pages 128–140.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Ting Xu, Huiyun Yang, Zhen Wu, Jiaze Chen, Fei Zhao, and Xinyu Dai. 2023. Measuring your aste models in the wild: A diversified multi-domain dataset for aspect sentiment triplet extraction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2837–2853.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Wenxuan Zhang, Yang Deng, Xin Li, Lidong Bing, and Wai Lam. 2021a. Aspect-based sentiment analysis in question answering forums. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4582–4591.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021b. Towards generative aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 504–510.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*.

Junxian Zhou, Haiqin Yang, Yuxuan He, Hao Mou, and JunBo Yang. 2023. A unified one-step solution for aspect sentiment quad prediction. *arXiv preprint arXiv:2306.04152*.

A Detailed Experimental Setup

We use AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of 1e-4 for training two T5 models. We set the batch size to 16 for training and 24 for inference. We train the first stage model for 30 epochs, and train 40 epochs for the second stage.

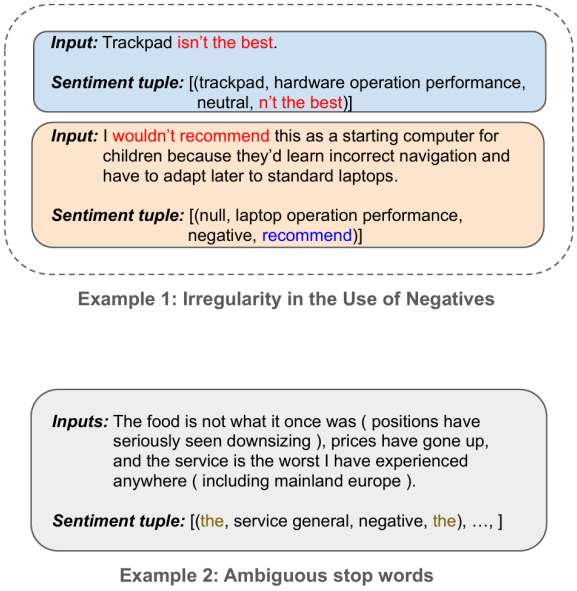


Figure 4: Two examples of irregularity of stop words. Note that these examples are the not all of the stop-word problems.

We also note that the dataset labels (i.e., the sentiment tuples) sporadically include stop words. For example, as in the first example of Figure 4, the inclusion of negations in the opinion terms is inconsistent. Also, as in the second example, element tuples sometimes contain ambiguous and meaningless stop words as elements. As a result, the fine-tuned model sometimes generates sentiment tuples containing stop words irregularly. It can yield critical performance degradation, even though they don't affect the meaning of the sentiment elements. To resolve the problem of stop words, we filter these stop words using nltk package(Farkiya et al., 2015) for both generated results and dataset labels. We use four RTX 4090 GPUs to train and evaluate all of the models.

B Case Study

We conduct a case study and analyze the properties of the outputs generated by the proposed method. As depicted in Figure 5, we classify the output results into three main cases.

The first case involves sentences that do not require multiple views for accurate prediction. For these sentences, our model succeeds in making efficient predictions using only a single view. We observe that this case is the most common type in our study, significantly contributing to the model's efficiency.

The second shows an example that predicts requires fewer views, but the example actually requires more views. Our analysis reveals that such cases frequently occurs with implicit O. As shown in Table 1, this suggests that our model's performance might lag behind other baselines on the ACOS Rest16 dataset, which contains many samples with implicit A and O. Additionally, the model struggles with predicting infrequent C in the training set. Incorporating the concept of self-information and defining the necessary number of views based on the 'amount of information in a sample' could effectively address this issue.

The final case involves cases with multiple sentiment tuples and longer lengths. We explain that errors in this scenario stem from two main reasons. Firstly, longer sentences include extended phrases that modify A or O. Including all these modifiers as elements often leads to errors, a common problem across different models that requires an alternative solution. Secondly, errors occur when the number of tuples is incorrectly predicted in the first stage. If the predicted number of tuples is insufficient, some target sentiment tuples might be overlooked. Conversely, overestimation leads to the extraction of irrelevant aspects, as depicted in the Figure 5. However, we optimize the first stage to reduce tuple count errors, which helped mitigate performance drops by minimizing incorrectly generated or overlooked tuples.

C Depth Analysis on First Stage

C.1 Accuracy on the Number of Views

We assess the accuracy of predicting the value of $\hat{K}$ and present the results in Table 4. We evaluate the output by comparing it to the number of labeled sentiment tuples using RMSE and accuracy. We carefully implement the first stage baselines to compare our method properly as follows: *Random*: We find that the number of sentiment tuples in the training dataset is mostly in the range of 1 to 6. For each inference, we randomly sample one of the 6 numbers and compare it with our first stage result. *Majority*: We also reveal that about

Case 1: Efficiency in SImple sentence

Input: Best mexican place for lunch in the financial district.

Target: [(mexican place, best, positive, restaurant general)]

Output: [(mexican place, best, positive, restaurant general)]

Case 2: One sentiment tuple, but complex

Input: The crowd is mixed yuppies, young and old.

Target: [(crowd, null, neutral, restaurant miscellaneous)]

Output: [(crowd, **mixed**, neutral, **ambience general**)]

Case 3: Complex sentence analysis

Input: If you ‘ re interested in good tasting (without the fish taste or smell), large portions and creative sushi dishes this is your place…

Target: [(null, good, positive, food quality), (portions, large, positive, food_style_options), (sushi dishes, creative, positive, food_style_options)]

Output: [(null, **good tasting**, positive, food quality), (portions, large, pos, food_style_options), (sushi dishes, creative, positive, food_style_options), (fish taste or smell, null, negative, food quality)]

Figure 5: Case study for the three main types of results. Blue one denotes correct, red one denotes incorrect, and the yellow one denotes irrelevant.

60 percent of labels consist of a single tuple. We construct a baseline that predicts only 1 for the number of tuples, to check whether our model has the ability to predict the number of sentiment tuples of a sentence. *Classification*: We adopt the RoBERTa model (Liu et al., 2019) to evaluate the results when treating the prediction of the number of views as a sequence classification task. We set the classes based on the number of sentiment tuples. As shown in Figure 6, the distribution of tuple counts is skewed towards the lower end, with instances containing more than seven tuples being nearly non-existent. Consequently, we limit the categories from 1 to 6 and clip instances with 7 or more tuples to 6. Additionally, to address label imbalance, we employ a weighted loss function, where the weights are set as the inverse of the frequency ratio for each category as in Equation (4). We use the same notation as in Section 2.1, and $\mathcal{I}()$ denotes the indicator function. This approach enables the model to effectively classify even the less represented classes.

$$\boldsymbol{W_c} = \frac{|D|}{\sum_D \mathcal{I}(min(|\boldsymbol{y}|, \boldsymbol{6}) = \boldsymbol{c})} \quad (c \in [1, 6])$$
$$\mathcal{L}_{cls} = -\sum_{i=1}^{|B|} \boldsymbol{W_{k_i}} \log p(\boldsymbol{k_i}|\boldsymbol{x_i}) \mathcal{I}(\boldsymbol{k_i} \leq \boldsymbol{6}) \tag{4}$$

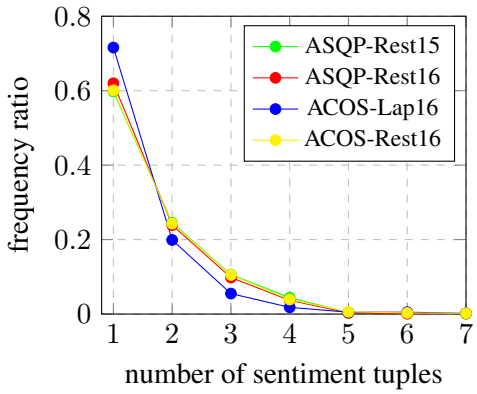


Figure 6: Distribution of the number of sentiment tuples. The sources are from training datasets of each task. We normalize each count by dividing it by the total number of data points. The number of tuples is clipped to 7.

C.2 Effect of Element Exclusions

We analyze the impact of excluding various marker tokens, including the [O] token representing opinions, to determine which token exclusions contribute to performance improvements. Additionally, we experiment with cases where no element exclusion is performed. In this section, we have also included the second stage results to provide a detailed comparison of the overall performance.

As in Table 4, our proposed method outperforms the other baselines and nearly predicts the actual distribution of sentiment tuples within a small mar-

Methods	First stage RMSE	First stage Acc.	Second stage F1 score
Random	2.80	18.89	-
Majority	0.99	63.39	-
Classification	0.83	61.90	-
DoT_{first}	0.54	77.83	**54.33**
exclude $[C]$	0.54	77.53	53.91
exclude $[A]$	**0.53**	77.77	53.71
exclude $[S]$	0.54	77.65	53.55
full elements	0.55	**78.22**	53.94

Table 4: First stage results for each main baseline and exclusion of specific tokens. We report average RMSE loss and accuracy for first stage, and F1 score for second stage.

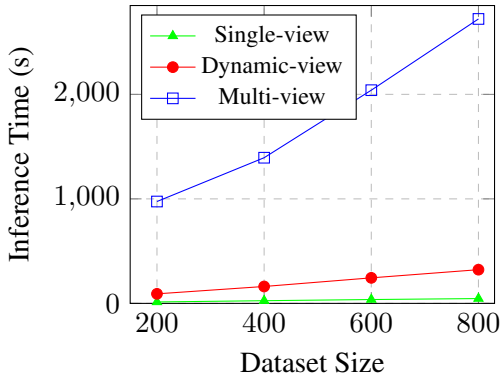


Figure 7: Inference time among dataset size for each model.

gin of error. This result justifies the use of the output from the first stage in the second stage. The first stage results in Table 4 do not exhibit significant performance differences among various exclusion. However, for the second stage results, which serve as the final output of this task, we observe a significant performance difference. The performance in the second stage is generally higher when O is omitted because generating O correctly is the most difficult and crucial task in quadruple prediction (Chebolu et al., 2023). If O is not trained in the first stage and is reused in the second stage, the model appears to focus more on learning about O compared to other elements, which already have some level of information.

D Computing Inference Time

We compare inference times based on view methods across different dataset sizes. The dataset consisted of randomly sampled test data from laptop16, with 200, 400, 600, and 800 samples. The baselines were set as static single view (T5-paraphrase) and static multi view (MvP), with the number of views for the multi view fixed at 15. Figure 7 shows that we not only dramatically reduce inference time of utilizing multi views, but also reduce the rate of increase in inference time with respect to the number of datasets. On the other hand, in terms of single view, we significantly increase F1 performance while suppressing the increase in inference time and the rate of its increase. These results suggest that the efficiency of our method becomes more pronounced as the dataset size increases.

E Input and Target Examples for Each Stage

In Figure 8, we provide detailed examples for input and output pairs in each stage. The input sentences in the dataset are presented in a basic sentence structure, while the labels consist of lists of sentiment tuples. To preprocess this data, during the first stage, the original input sentence is kept unchanged, and the target is set as the initial order template, which consisted of a number of views corresponding to the number of sentiment tuples in the label. In the second stage, the input is processed by appending the final order template as a prompt to the original input sentence. The target is then constructed by adjusting the order of the elements within the sentiment tuples to align with the corresponding views in the order template.

F Analysis on Implicit Term

In Table 1, DOT suffers from predicting sentiment tuples in Rest and Clothing domains. We noted that the ACOS dataset contains a significant number of instances with implicit aspects or implicit opinions. Additionally, we discovered that the Rest and Clothing dataset are smaller in scale compared to other ACOS datasets. The scale of each dataset and the number of instances containing implicit terms are recorded in Table 5. Based on these observations, we hypothesized that the size of the dataset and the distribution of implicit terms contribute to the performance degradation observed in the Rest and Clothing datasets.

As shown in Table 6, it is evident that the F1 score for instances containing implicit terms in the Rest dataset is significantly lower compared to using the paraphrase method. Additionally, we

Original input: Helpful service and average price per dish $10.
Original target: [(service, service general, positive, helpful), (dish, food prices, neutral, $10)]

First stage input: Helpful service and average price per dish $10.
First stage target: *[S] [A] [C] [SSEP] [A] [C] [S]*

Second stage input: Helpful service and average price per dish $ 10. *[S] [A] [O] [C] [SSEP] [A] [O] [C] [S]*
Second stage target: *[S]* positive *[A]* service *[O]* helpful *[C]* service general *[SSEP] [A]* dish *[O]* $ 10 *[C]* food prices *[S]* neutral

Figure 8: Examples for input and target from original dataset for both first and second stage.

Datasets	ASQP		ACOS		MEMD				
	R15	R16	Lap	Rest	M-Rest	M-Laptop	Books	Clothing	Hotel
total samples	834	1264	2934	1530	3622	2863	2092	1674	2481
implicit samples	272	446	1826	822	1801	1751	1523	1083	1278
implicit sample %	32.6	35.3	62.2	53.7	49.7	61.2	72.8	64.7	51.5

Table 5: The size of each dataset and the number of samples containing implicit terms. For ease of comparison, We also provide the percentage of samples with implicit terms relative to the total number of samples. It is evident that the implicit term ratio in the ACOS dataset is higher compared to that in the ASQP dataset.

Methods	Rest	M-Rest ¼	M-Rest ½	M-Rest full
Paraphrase	**50.06**	**40.26**	47.77	49.09
DOT	44.84	35.74	**47.81**	**49.49**

Table 6: F1 scores only for samples containing implicit terms. We report the performance in Rest dataset and the performance trends across different dataset scales.

observed a performance degradation when training on a randomly selected quarter of the M-Rest dataset. However, as the amount of training data from the M-Rest dataset increased, the performance on implicit terms improved, eventually surpassing the F1 score of the paraphrase method in the full M-Rest dataset. This result demonstrates that the small size of the dataset with a high proportion of implicit terms is the primary cause of the performance degradation in the Rest and Clothing dataset. It also suggests that the performance is likely to improve as the dataset size increases.

G Additional Analysis

In this section, we conduct an in-depth analysis of various aspects of our model. For a comprehensive evaluation, we use Paraphrase and MvP as baselines, running identical experiments for comparison. We assess performance across multiple tasks using several benchmarks, including R15, R16, Lap, Rest, and M-Rest.

Different Backbone Model We conduct the experiment using different encoder-decoder based model, BART (Lewis, 2019) as backbone model. We utilize BART with the same hyperparameters and data processing techniques applied to the T5 model for three methods including ours. However, as in Table 7, we observe a noticeable decline in overall F1-scores for all models, primarily due to insufficient hyperparameter tuning compared to T5. Nevertheless, as shown in the results, our method still outperforms the baseline models with BART, suggesting that its effectiveness is not highly dependent on the choice of backbone model.

Methods	ASQP		ACOS		
	R15	R16	Lap	Rest	M-Rest
Paraphrase	31.77	38.15	30.98	36.65	35.74
MvP	33.48	41.01	32.57	**40.40**	40.30
DOT	**35.98**	**41.73**	**33.12**	39.61	**40.91**

Table 7: F1 score on benchmark datasets using BART as the backbone model.

Complex Sentences As mentioned in Appendix B, processing long and complex contexts is a well-known challenge, and our model performs similarly to others in this regard. We define complex sentences as those containing more than three sentiment tuples, exceeding 22 words in length, or having a Flesch-Kincaid Grade Level (Solnyshkina et al., 2017) of 9 or higher, representing the top 20% for each criterion. We sample these complex sentences and evaluate the F1 scores for these samples. As shown in the Table 8, performance degradation in complex sentences is a common issue across all

models. We attribute the larger performance drop in our model compared to MvP to the fact that it uses fewer views, which limits its capacity to thoroughly analyze complex sentences. Nevertheless, our model's final Complex F1 score remains close to that of MvP and surpasses that of the Paraphrase model.

Methods	ASQP		ACOS		
	R15	R16	Lap	Rest	M-Rest
Paraphrase	44.94	55.73	37.15	**56.12**	54.37
MvP	46.71	**58.00**	37.68	56.06	**58.23**
DOT	**46.93**	57.70	**38.57**	54.16	57.72

Table 8: F1 scores evaluated on complex samples only.

Training Complexity Our method may appear complex due to the numerous components that require training. However, since our method involves simply training the T5 model twice without complex optimization procedures, the overall training time is not significantly longer than that of other models. As shown in the Table 9, even though our model uses 30 and 40 epochs for two stages of training—more than the 20 epochs used in MvP—the total training time remains much shorter than that of MvP. In terms of memory usage, only two T5 models are allocated in memory, so the memory consumption does not increase exponentially compared to existing models.

Methods	ASQP		ACOS		
	R15	R16	Lap	Rest	M-Rest
Paraphrase	212.83	314.18	652.31	349.79	815.48
MvP	3883.74	5008.84	11006.07	6169.02	14634.71
DOT	1161.73	1648.61	3310.63	1814.41	4157.93

Table 9: Training duration for each benchmark.

Standard Deviation We conduct experiments using five different random seeds and calculate the standard deviation of the outcomes. Results are reported in Table 10. Our findings indicate that our model exhibits a higher overall standard deviation compared to other baselines. This can be attributed to the structure of the method, where an error at one stage is likely to propagate and accumulate. However, it is important to note that the absolute value of the standard deviation is not significantly large. In fact, the higher variation suggests that the model may possess greater potential to achieve stronger performance.

Methods	ASQP		ACOS		
	R15	R16	Lap	Rest	M-Rest
Paraphrase	± 0.44	± 0.64	± 0.26	± 0.68	± 0.38
MvP	± 0.54	± 0.29	± 0.48	± 0.72	± 0.48
DOT	± 0.74	± 0.85	± 1.01	± 0.76	± 0.42

Table 10: Standard deviation of outcomes in Table 1.

H Detailed Setups for LLM Experiments

As in Table 1, we perform the ABSA task using the GPT-4o, LLaMa-3.1-8B, and Mistral-7B models, compairing the results with our DOT model. For the GPT model, we utilize in-context learning (Brown et al., 2020). We randomly sample 10 instances and combine them with instruction format, and add it as a prompt. For the other three open-source LLMs, we employ instruction tuning (Wei et al., 2021) with the training dataset for fine-tuning, using the same instructions as in GPT prompts. To ensure stable model training during fine-tuning, we utilize the LoRa (Hu et al., 2021). We present the specific prompts and framework in Figure 9.

```
According to the following sentiment elements definition:

- The 'aspect term' refers to a specific feature, attribute, or aspect of a product
    or service that a user may express an opinion about, the aspect term might be '
    null' for implicit aspect.
- The 'opinion term' refers to the sentiment or attitude expressed by a user towards
     a particular aspect or feature of a product or service, the aspect term might
    be 'null' for implicit opinion.
- The 'aspect category' refers to the category that aspect belongs to, and the
    available categories includes: {dataset specific categories}.
- The 'sentiment polarity' refers to the degree of positivity, negativity or
    neutrality expressed in the opinion towards a particular aspect or feature of a
    product or service, and the available polarities inlcudes: 'positive', 'negative
    ' and 'neutral'.

Recognize all sentiment elements with their corresponding aspect terms, aspect
    categories, opinion terms and sentiment polarity in the following text with the
    format of [('aspect term', 'aspect category', 'sentiment polarity', 'opinion
    term'), ...]:
```

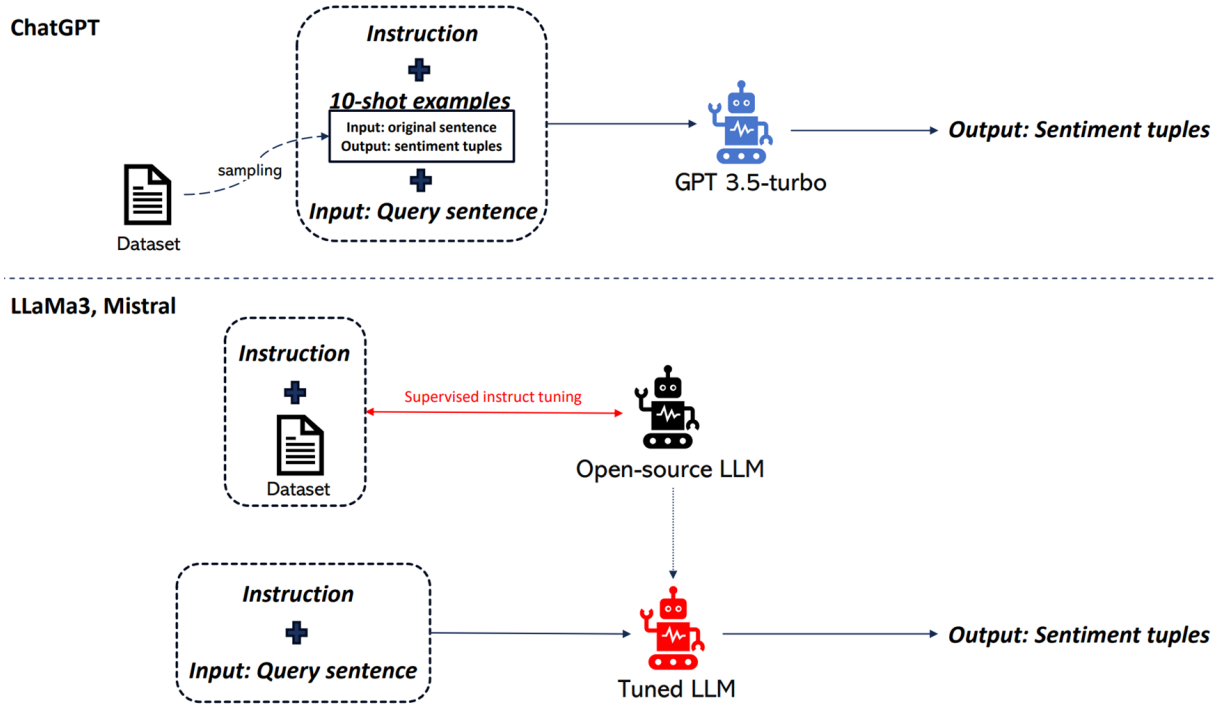


Figure 9: Instruction format for two LLM frameworks. We utilize in-context learning for GPT-3.5-turbo inference, and instruction-tuning for LLaMa-3.1 and Mistral inference respectively.

That doesn't sound right: Evaluating speech transcription quality in field linguistics corpora

Éric Le Ferrand[1], **Bo Jiang**[1], **Joshua Hartshorne**[2], **Emily Prud'hommeaux**[1]
[1]Department of Computer Science, Boston College, USA
[2]MGH Institute of Health Professions, USA

Abstract

Incorporating automatic speech recognition (ASR) into field linguistics workflows for language documentation has become increasingly common. While ASR performance has seen improvements in low-resource settings, obstacles remain when training models on data collected by documentary linguists. One notable challenge lies in the way that this data is curated. ASR datasets built from spontaneous speech are typically recorded in consistent settings and transcribed by native speakers following a set of well designed guidelines. In contrast, field linguists collect data in whatever format it is delivered by their language consultants and transcribe it as best they can given their language skills and the quality of the recording. This approach to data curation, while valuable for linguistic research, does not always align with the standards required for training robust ASR models. In this paper, we explore methods for identifying speech transcriptions in fieldwork data that may be unsuitable for training ASR models. We focus on two complimentary automated measures of transcription quality that can be used to identify transcripts with characteristics that are common in field data but could be detrimental to ASR training. We show that one of the metrics is highly effective at retrieving these types of transcriptions. Additionally, we find that filtering datasets using this metric of transcription quality reduces WER both in controlled experiments using simulated fieldwork with artificially corrupted data and in real fieldwork corpora.

1 Introduction

Automatic speech recognition (ASR) can support the creation of new linguistic resources for under-resourced and endangered languages, but such languages – which make up the vast majority of the world's 7000+ languages – vary considerably in their quantity of transcribed speech data. While some languages have well-curated speech datasets sourced from educational materials, mass media, or crowdsourcing efforts, many only have field recordings made by linguists as their primary source of speech data. Field linguists, whose work centers on describing languages and analyzing their linguistic properties, collect data primarily to support their academic research and the activities of the language community. Few linguists collect data with the goal of creating high-quality datasets for training speech technology models (Hanke, 2017; Le Ferrand, 2023). As a result, speech data from fieldwork may be only partially or unfaithfully transcribed due to issues of recording quality and the language skills of the linguist. Additionally, fieldwork transcripts often include ancillary information making it difficult to differentiate between transcription (word-level renderings of the speech) and annotation (glosses, translations, comments).

ASR models for widely-spoken languages are trained on enough professionally recorded and transcribed data that including a small number of inaccurately transcribed utterances is unlikely to significantly affect overall performance. For languages where data is scarce, however, even a small portion of low-quality data can severely degrade model performance. Detecting low quality transcripts can be done manually, but this process is tedious and time-consuming, underscoring the need for an automatic method to evaluate transcription quality.

In this paper, we explore two metrics for automatically assessing the transcription quality and accuracy of speech datasets: Phonetic Distance Match (PDM), a novel metric based on phoneme recognition, and the posterior probability of a Connectionist Temporal Classification (Graves and Graves, 2012) alignment (CTC). We evaluate the utility of these metrics for identifying poor transcriptions through experiments on clean datasets that we synthetically corrupt in ways that simulate common fieldwork data quality errors. We then demonstrate the ability of PDM in particular to

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 627–635
July 27 - August 1, 2025

identify these types of errors. Finally we show that using our metrics to filter out these types of inaccurate transcripts can yield substantial improvements in ASR accuracy in both simulated and real-world fieldwork datasets.

2 Related work

Prior related work on filtering inaccurate transcripts has focused on leveraging ASR output itself, such as using the confidence score of an ASR model (Huang et al., 2013; Su and Xu, 2015; Koctúr et al., 2016) or using multiple ASR models to generate multiple predictions for the same speech utterance (Fiscus, 1997; Cucu et al., 2014; Li et al., 2016; Jouvet and Fohr, 2014). These methods, while useful in high-resource settings, assume the existence of an ASR model for the target language, which is not applicable for our work, where no strong ASR models exist. Using universal speech recognition models shows more promise for under-resourced languages. Models such as XLS-R (Conneau et al., 2021) and MMS (Pratap et al., 2024) have demonstrated promising ASR results in low-resource settings (Macaire et al., 2022; Guillaume et al., 2022; Tapo et al., 2024; Romero et al., 2024; Jimerson et al., 2023). However, since these models are trained on raw speech and lack textual information, they cannot provide direct feedback on individual segments. A more suitable alternative is a universal phoneme recognizer (Li et al., 2020), which can generate phone transcriptions for any language.

There is a robust history of prior work in evaluating the acoustic quality of audio using measures of speech intelligibility derived from output of ASR or proto-ASR systems (Holube and Kollmeier, 1996; Sakoe and Chiba, 1978; Spille et al., 2018; Arai et al., 2019). While this work is also relevant for filtering audio for ASR datasets, it is orthogonal to our own work, which focuses on identifying poor quality transcripts rather than poor quality audio.

3 Data

We apply our metrics (see Section 4.1) to two distinct classes of datasets. The CURATED class consists of five well-curated, high-quality speech datasets, ranging in size from 3.5h to 9h, which we synthetically corrupt to simulate common fieldwork transcription errors (see Section 4.2). The languages include Bunun (bnn), Saisiyat (xsy), and Seediq (trv), three Taiwanese indigenous languages extracted from FormosanBank (Mohamed et al., 2024). For each language, we use a subset of the ePark (Aboriginal Language Research and Development Foundation, 2023b) and ILRDF (Aboriginal Language Research and Development Foundation, 2023a) corpora, which consist of read speech recorded by native speakers (Hartshorne et al., 2024). We also included Mboshi (mdw), a Bantu language from Congo-Brazzaville, part of the LIG-Aikuma project[1], and Duoxu (ers), a critically endangered Sino-Tibetan language, included in the Pangloss collection (Michailovsky et al., 2014).

The second class consists of 2-hour fieldwork corpora from Pangloss (FIELDWORK) for Namakura (nmk), an Austronesian language spoken on Vanuatu, and Thulung Rai (tdh), a Sino-Tibetan language of Nepal. These consist exclusively of fieldwork recordings and include annotations and approximate transcripts. We use the FIELDWORK datasets to demonstrate the efficacy of our methods in a real-world fieldwork scenario.

All seven datasets[2] were partitioned into training (70%), validation (10%), and test (20%) sets. Dataset details are found in Table 1.

Several factors motivated our choice of these specific languages. First, the FormosanBank corpus contains an unusually large amount of high-quality data. These languages also posed an initial layer of complexity due to their orthographic conventions. For example, the glottal stop is typically represented with an apostrophe or straight single quote, and the voiceless alveolar affricate is denoted as *c*. Mboshi includes two non-ASCII characters—ɛ and ω—while Duoxu features systematic tone marking using superscript numerals (e.g., $ja^{22}nje^{33}\ xe^{53}nje^{33}\ tɕi^{33}\ o$), adding another dimension of orthographic variation.

4 Method

4.1 Transcript evaluation metrics

We consider two metrics for evaluating transcription quality[3]. First, we present **Phonetic Distance Match (PDM)**, a novel metric for evaluating orthographic transcriptions against their corresponding audio. PDM is calculated by transcribing an utterance recording using a phone-level transcription model and then measuring the edit distance between the resulting transcription and the manual reference transcription. Using Allosaurus (Li

[1] github.com/besacier/mboshi-french-parallel-corpus
[2] https://github.com/eleferrand/data_quality_eval
[3] https://github.com/eleferrand/data_quality_eval

et al., 2020) without fine-tuning, we automatically generate phone-level transcripts for each utterance in the corpus, which are then converted into their closest corresponding ASCII characters using the `unidecode` library.[4] The orthographic reference transcripts are also converted to ASCII to ensure a shared character set. Finally, we compute the normalized Levenshtein distance between the two transcriptions and subtract from 1 to generate a similarity metric ranging from 0 to 1. The scoring process is illustrated in detail in Appendix Fig. 4.

The rationale behind using an ASCII-ized version of IPA is as follows. We begin with the observation that many languages currently being documented are traditionally oral. Their orthographies are often introduced by outsiders who tend adopt the Latin alphabet, with minor modifications. Although exceptions exist (e.g., Ainu written in Japanese katakana or Inuktitut written using Indigenous Canadian syllabics), the Latin script remains the prevalent standard.

While we recognize that linguists and community members who use the Latin alphabet are free to use the characters as they wish in their writing systems, these newly devised orthographies are not arbitrary. They are frequently influenced by existing Latin-based writing systems and the International Phonetic Alphabet (IPA). For example, a voiced velar nasal is typically represented as ŋ or *ng*, and rarely as unrelated letters like *p* or *r*. Naturally, inconsistencies can occur—such as c representing /s/, /k/, or /ʃ/ in French, or /ts/ in Seediq, but overall, we expect the ASCII-ized forms to retain at least phonemic consistency.

There are two advantages to our approach. First, it does not require any prior knowledge about the language or its phonetic inventory, which might not be easily available for a poorly documented language. Second, it does not require additional effort or resources to create a rule-based or learned G2P transformation of the data. In short, the method can be applied to any language that uses at least a partially ASCII-based transcription system without requiring additional model training or in-depth research into the phonetic properties of the language.

The second metric is the **Connectionist Temporal Classification (CTC)** alignment posterior probability. We use a large wav2vec (Baevski et al., 2020) model[5] to extract a speech representation from each utterance, again without fine-tuning; we then apply CTC alignment (Graves et al., 2006) between the speech features and the manual transcription and output the alignment posterior probability. It is entirely independent of the PDM metric.

4.2 Synthetic dataset corruption

To simulate a dataset containing typical fieldwork transcription errors, we arbitrarily select 20% of each training set of the 5 CURATED datasets and introduce transcription errors using three different corruption methods: (1) **Deleted**: three random words are removed from the transcription; (2) **Cropped**: the final 50% of words in the transcription are removed; (3) **Swapped**: the transcription is randomly replaced with another from the training set. For each language, we create three corrupted datasets, each containing 20% of the utterances corrupted in one of these three ways. Each utterance/transcript pair in the three datasets is then scored with the two metrics described in Sec. 4.1. We then evaluate how accurate our metrics identified these corrupted utterances. Examples of corrupted utterances can be found in Table 2.

4.3 ASR model building

All experiments are conducted with XLSR-53 (Conneau et al., 2021), a multilingual model based on the wav2vec architecture. We train a CTC layer for 30 epochs, selecting the best model with the validation set. Decoding is performed using a trigram LM trained on the training set for each language and corruption setting. We follow the popular XLS-R tutorial[6] but do not freeze the feature extractor.

In our **simulated scenario**, we use the three corrupted versions of each CURATED dataset (cf. Sec. 4.2). For each corrupted dataset, as well as for the uncorrupted dataset, we train an ASR model to determine the impact of each corruption on WER. For each corrupted dataset, we then create three filtered datasets: one in which we filter out 20% of the utterances according to the strength of the PDM metric; one where we do the same according to the CTC metric. When training data is limited, removing utterances from the training set can negatively impact performance. To ensure a fair comparison, we also evaluate performance using a dataset where the same percentage of utterances is removed from the training data at random.

In our **real-world scenario**, we calculate the two

[4] https://pypi.org/project/Unidecode

[5] https://huggingface.co/facebook/wav2vec2-base-960h

[6] https://huggingface.co/blog/fine-tune-xlsr-wav2vec2

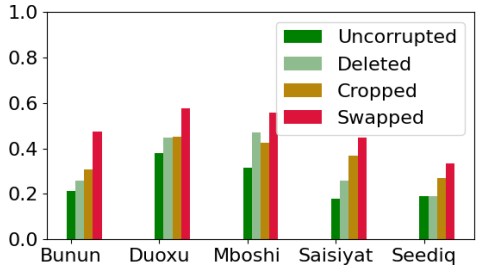


Figure 1: WER across corruption configurations.

metrics on the utterances of the two FIELDWORK datasets. For each dataset, we train ASR models on the unfiltered dataset and on filtered datasets, removing 5%, 10%, and 20% of the utterances using the more promising PDM metric and via random selection.

5 Results

5.1 Detecting corrupted transcripts

Figure 5 shows the full ROC curves and AUC values for all combinations of corruption type, dataset, and metric. We see that PDM achieves near-perfect AUC scores (0.89-0.98) for detecting utterances in the Swapped configuration, very high scores for Cropped (0.77-0.94), and strong scores for Deleted (0.64-0.85). In contrast, CTC is consistently and substantially less effective for all languages in all three corruption settings, with some AUC scores performing at chance in the Deleted and Cropped setting. Notably, Duoxu and Mboshi exhibit lower AUCs perhaps due to weak overlap in character set with the English wav2vec model used (cf. Table 1). The Deleted configuration appears to be the most challenging to detect for both metrics, but with PDM showing a clear advantage over CTC.

5.2 ASR evaluation: Simulated fieldwork

The baseline results for both the uncorrupted and corrupted CURATED datasets, shown in Figure 1, reveal a clear trend. The Deleted configuration causes the least degradation in WER. The Cropped configuration generally yields the second-worst results, except for Mboshi, where Deleted performs worse perhaps because of Mboshi's shorter average utterance length. Finally, the Swapped configuration consistently produces the weakest WER.

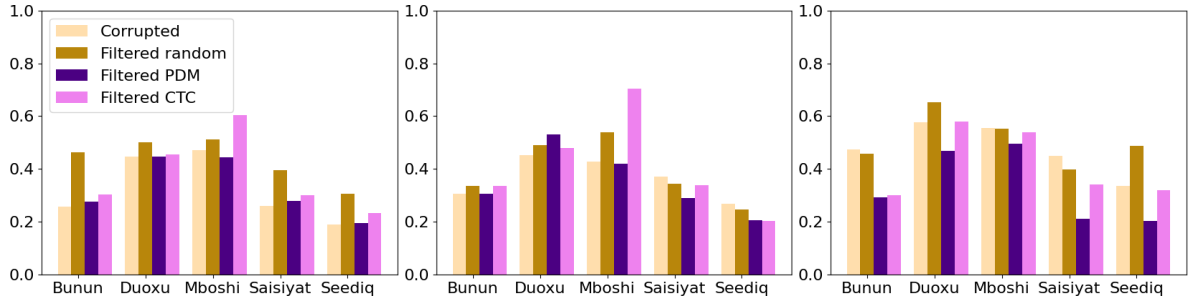
Figure 2 shows changes in WER in the corrupted datasets with and without filtering using the two metrics, PDM and CTC, as well as the random setting where 20% of the data is removed from a corrupted dataset at random. In the Deleted setting, PDM filtering has minimal impact, while CTC filtering generally degrades WER. In the Cropped setting, filtering with PDM improves WER except for Duoxu, while CTC filtering again generally degrades WER. In the Swapped setting, filtering with PDM systematically and often dramatically improves WER, while CTC filtering has little impact except for Bunun and Saisiyat. Overall, the superior performance of PDM filtering is quite consistent, yielding better results than CTC filtering in 14 out of 15 cases. The exception is Duoxu in the Cropped setting. As already noted, Duoxu's writing system contains many non-ASCII characters, which may limit the performance of the PDM metric in some cases.

In a few rare cases, ASR models trained on a corrupted dataset outperform those trained on data filtered using one of the two metrics. This typically occurs when the filtering metric lacks sufficient accuracy, as is observed in some languages with the Deleted and Cropped configurations (see Figure 5), leading to the unintended removal of clean data while allowing corrupted data to remain, ultimately degrading performance. The Deleted setting, which has a minor impact when utterances are relatively long, may also serve unintentionally as a form of corruption-based regularization.

5.3 ASR evaluation: Real-world fieldwork

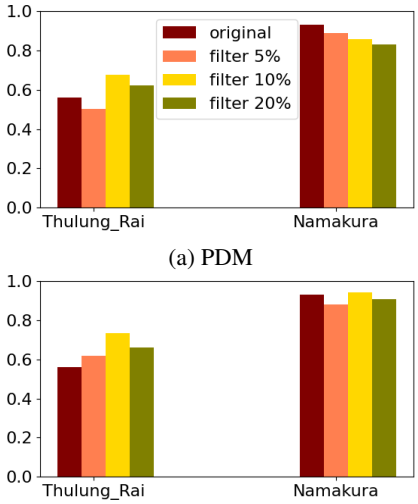
Figure 3 shows the results of different thresholds of PDM filtering and random filtering on the two FIELDWORK real-world datasets. (We do not report results for CTC given the weak utility observed in the simulated fieldwork scenario both for corruption detection and as a filter.) For Thulung Rai, a 5% filtering threshold proved the most effective, resulting in a decrease of several points in WER, while higher thresholds and random filtering resulted in WER increases. With the Namakura dataset, WER consistently decreased as more data was filtered using the PDM score, suggesting that a significant portion of the corpus may contain transcription errors. Filtering randomly for Namakura yielded slight random variations in WER.

To better understand the utility of the PDM metric for identifying poor transcripts, we manually inspected the transcriptions of the lowest and highest 5% of utterances based on PDM scores for both corpora. In Thulung Rai, 61% of the lowest scoring utterances showed no issues, while 14% had mismatched transcriptions and 23% contained cropped transcriptions. In contrast, 93% of the top scoring



(a) Deleted (b) Cropped (c) Swapped

Figure 2: WER for corrupted and filtered CURATED datasets in the simulated fieldwork scenario.



(a) PDM

(b) Random

Figure 3: WER for unfiltered and filtered FIELDWORK datasets in the real-world fieldwork scenario.

utterances had correct transcriptions, with 6% missing a few words. For Namakura, only 11% of the lowest scoring utterances had accurate transcriptions, with 55% mismatched, 29% cropped, 3% with cropped audio, and 1% missing some words. Conversely, the highest scoring utterances had 97% correct transcriptions, with 1.5% cropped and another 1.5% missing words.

6 Conclusions and Future Work

This paper explores two metrics for identifying unsuitable and inaccurate speech transcriptions to improve ASR training from linguistic fieldwork data with the goal of supporting language documentation. We find that our novel PDM metric and, to a lesser extent, a CTC confidence metric are effective in identifying erroneous transcriptions in both simulated and real-world fieldwork datasets. Moreover, filtering data using the PDM metric consistently reduces WER in both simulated and real-world fieldwork scenarios. In our future work, we plan to investigate additional methods for identifying poor transcriptions and to explore the relationship between audio quality and transcription quality.

Limitations

Experimental results demonstrate that our PDM method is highly effective for languages with a limited number of non-ASCII characters. However, further experiments are needed to evaluate its performance on languages with a larger set of non-ASCII characters and non-Latin writing systems. The proposed metrics efficiently identify major errors, such as missing or mismatched transcripts, but are less likely to detect spelling mistakes or inconsistent transcription of specific speech sounds, which could also significantly impact WER. While these methods could be applied to high-resource languages like French or German, such languages may benefit more from approaches leveraging existing G2P models or pre-trained ASR systems trained specifically for these languages.

Ethics Statement

Researchers must always be respectful of language community concerns about data ownership when working with Indigenous language data. All of our data is gathered from public sources. In the case of the Formosan languages, the two organizations providing the data, the Indigenous Languages Research and Development Foundation and the ePark educational research organization, actively seek out collaborations with computational researchers. The other datasets are also made available on the Web by their creators specifically with the goal of furthering research in these languages, both linguistic and computational. We have permission from the creators and owners to redistribute the data in the form of ASR datasets.

Acknowledgments

The authors thank Yuyang Liu, Li-May Sung, and the Indigenous Languages Research and Development Foundation, especially Akiw and Lowking Nowbucyang, for generously providing data. This material is based upon work supported by the National Science Foundation under Grant #2319296. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Aboriginal Language Research and Development Foundation. 2023a. Online dictionary of aboriginal languages. https://e-dictionary.ilrdf.org.tw.

Aboriginal Language Research and Development Foundation. 2023b. yuanzhumin yuyan leyuan (epark). https://web.klokah.tw/.

Kenichi Arai, Shoko Araki, Atsunori Ogawa, Keisuke Kinoshita, Tomohiro Nakatani, Katsuhiko Yamamoto, and Toshio Irino. 2019. Predicting speech intelligibility of enhanced speech using phone accuracy of DNN-based ASR system. In *Interspeech*, pages 4275–4279.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2021. Unsupervised cross-lingual representation learning for speech recognition. *Interspeech 2021*.

Horia Cucu, Andi Buzo, and Corneliu Burileanu. 2014. Unsupervised acoustic model training using multiple seed asr systems. In *Spoken Language Technologies for Under-Resourced Languages*.

Jonathan G Fiscus. 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 347–354. IEEE.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Alex Graves and Alex Graves. 2012. Connectionist temporal classification. *Supervised sequence labelling with recurrent neural networks*, pages 61–93.

Séverine Guillaume, Guillaume Wisniewski, Benjamin Galliot, Minh-Châu Nguyên, Maxime Fily, Guillaume Jacques, and Alexis Michaud. 2022. Plugging a neural phoneme recognizer into a simple language model: a workflow for low-resource setting. In *Proceedings of Interspeech*, pages 4905–4909.

Florian Hanke. 2017. *Computer-Supported Cooperative Language Documentation*. Ph.D. thesis, Ph. D. thesis, University of Melbourne.

Joshua K. Hartshorne, Éric Le Ferrand, Li-May Sung, and Emily Prud'hommeaux. 2024. Formosanbank and why you should use it. In *Architectures and Mechanisms in Language Processing (AMLaP) Poster*.

Inga Holube and Birger Kollmeier. 1996. Speech intelligibility prediction in hearing-impaired listeners based on a psychoacoustically motivated perception model. *The Journal of the Acoustical Society of America*, 100(3):1703–1716.

Yan Huang, Dong Yu, Yifan Gong, and Chaojun Liu. 2013. Semi-supervised GMM and DNN acoustic model training with multi-system combination and confidence re-calibration. In *Interspeech*, pages 2360–2364.

Robert Jimerson, Zoey Liu, and Emily Prud'hommeaux. 2023. An (unhelpful) guide to selecting the best ASR architecture for your under-resourced language. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1008–1016.

Denis Jouvet and Dominique Fohr. 2014. About combining forward and backward-based decoders for selecting data for unsupervised training of acoustic models. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association*.

Tomáš Koctúr, Ján Staš, and Jozef Juhár. 2016. Unsupervised acoustic corpora building based on variable confidence measure thresholding. In *2016 International Symposium ELMAR*, pages 31–34. IEEE.

Éric Le Ferrand. 2023. *Leveraging Speech Recognition for Interactive Transcription in Australian Aboriginal Communities*. Ph.D. thesis, Charles Darwin University.

Sheng Li, Yuya Akita, and Tatsuya Kawahara. 2016. Data selection from multiple ASR systems' hypotheses for unsupervised acoustic model training. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5875–5879. IEEE.

Xinjian Li, Siddharth Dalmia, Juncheng Li, Matthew Lee, Patrick Littell, Jiali Yao, Antonios Anastasopoulos, David R Mortensen, Graham Neubig, Alan W Black, et al. 2020. Universal phone recognition with a multilingual allophone system. In *ICASSP 2020-2020 IEEE International Conference on Acoustics,*

Speech and Signal Processing (ICASSP), pages 8249–8253. IEEE.

Cécile Macaire, Didier Schwab, Benjamin Lecouteux, and Emmanuel Schang. 2022. Automatic speech recognition and query by example for Creole languages documentation. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Boyd Michailovsky, Martine Mazaudon, Alexis Michaud, Séverine Guillaume, Alexandre François, and Evangelia Adamou. 2014. Documenting and researching endangered languages: The Pangloss Collection. *Language Documentation and Conservation*, 8:119–135.

Wael Mohamed, Éric Le Ferrand, Li-May Sung, Emily Prud'hommeaux, and Joshua Hartshorne. 2024. Formosanbank. https://ai4commsci.gitbook.io/formosanbank.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, et al. 2024. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52.

Monica Romero, Sandra Gómez-Canaval, and Ivan G Torre. 2024. Automatic speech recognition advancements for Indigenous languages of the Americas. *Applied Sciences*, 14(15):6497.

Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

Constantin Spille, Stephan D Ewert, Birger Kollmeier, and Bernd T Meyer. 2018. Predicting speech intelligibility with deep neural networks. *Computer Speech & Language*, 48:51–66.

H Su and H Xu. 2015. Multi-softmax deep neural network for semi-supervised training. In *Proceedings of Interspeech*, pages 3239–3243.

Allahsera Tapo, Éric Le Ferrand, Zoey Liu, Christopher Homan, and Emily Prud'hommeaux. 2024. Leveraging speech data diversity to document Indigenous heritage and culture. In *Proceedings of Interspeech 2024*, pages 5088–5092.

A Appendix

Table 1 shows the durations, token and type counts, and non-ASCII character proportions of each of the 7 datasets for reference purposes. As noted in the paper, we have released these corpora and their partitions for research purposes. We note that they are derived in their entirety from publicly available sources with licensing that permits redistribution in other formats.

Table 2 provides examples of the three types of corruption designed to mimic the kinds of errors observed in fieldwork transcripts.

Figure 2 provides a walk-through of the PDM calculation process with three example utterances.

Figure 5 plots all six ROC curves and reports AUC measures for using each of the two metrics, PDM and CTC, to identify corruptions for each CURATED dataset under of the three corruption settings.

Table 3 shows the WER results presented in the paper in graphical format in Figure 2.

	Bunun	Duoxu	Mboshi	Saisiyat	Seediq	Namakura	Thulung Rai
Duration	8h34	7h57	3h28	8h14	8h53	1h53	2h18
Token	40166	61564	25671	39644	50123	18566	16296
Type	6846	2557	5621	4723	5608	1065	3965
Non-ASCII	0%	44%	23%	0%	2%	1%	8%

Table 1: Corpus size and token/type count for all datasets. We also provide the percentage of non-ASCII characters which may have an impact on the utility of the PDM metric.

Config.	Original	Corrupted
Deleted	wa adi pósá báabará wa kaá kobhá epωrωrω baá óyáála mwána anyωω	wa pósá wa epωrωrω óyáála mwána
Cropped	maqasmav a abus malaitaz a savi to seediq msgelu sa seediq mneyah alang kiya	maqasmav a abus to seediq msgelu sa
Swapped	supah a samah sia humacia tai hari niqan rebuq watan dao su trebuq hii	anak anak sa ia maupacia minhanglas slii hini kanna nnapa namu bunga

Table 2: Examples of input utterances and their corruptions from the three corruption configurations.

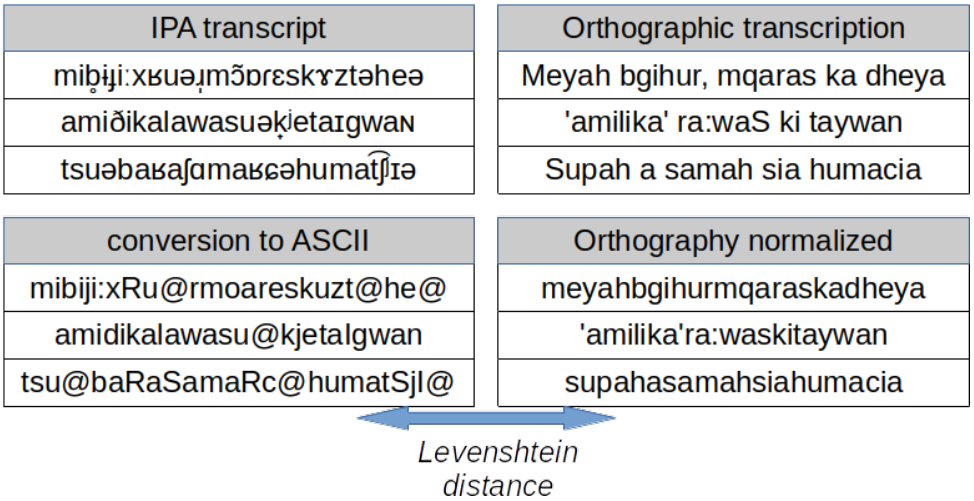
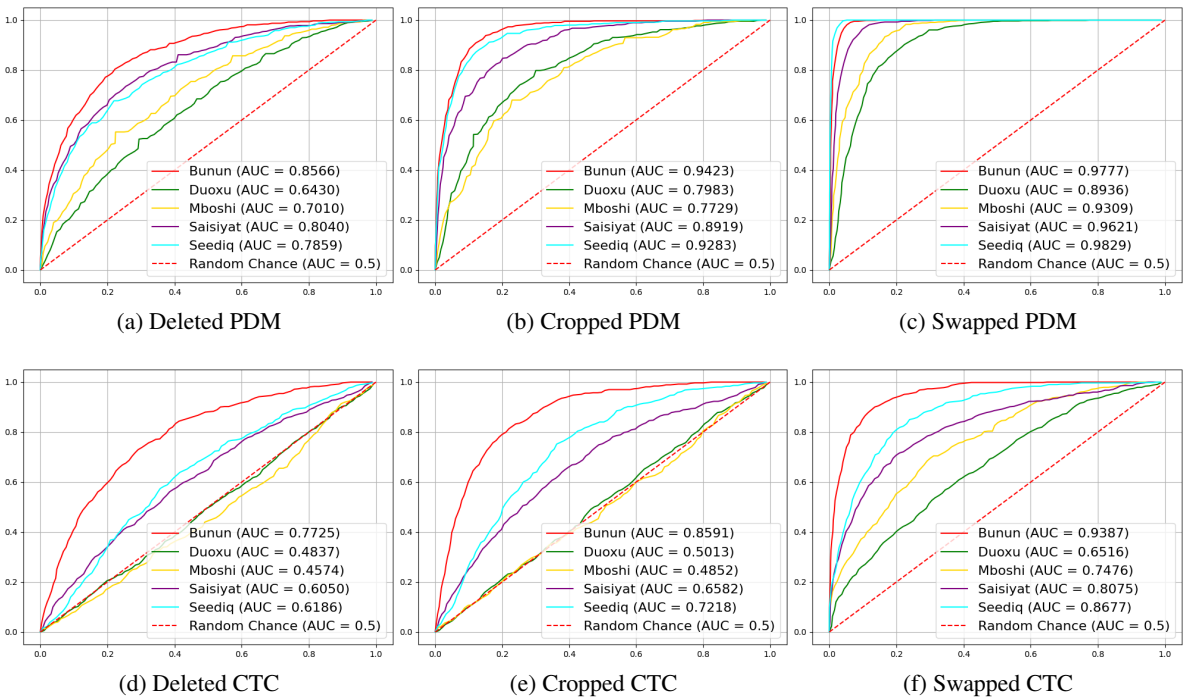


Figure 4: Demonstration of the PDM calculation method. In the upper left we see IPA transcripts generated from audio by Allosaurus. In the upper right we see the corresponding reference orthographic transcription for the three sample utterances. In the lower left are the phone-level transcripts converted to their ASCII equivalents often used to represent those IPA symbols (e.g., with Sampa). In the lower right, the reference orthographic transcriptions converted to ASCII, with spaces removed. We calculate normalized Levenshtein distance between the utterance in the lower panels and subtract from 1 to create the PDM metric.



(a) Deleted PDM (b) Cropped PDM (c) Swapped PDM

(d) Deleted CTC (e) Cropped CTC (f) Swapped CTC

Figure 5: ROC curves comparing performance of PDM and CTC for retrieving corrupted trasncriptions under the three corruption settings for all five of the CURATED datasets.

Corruption Setting	Filtering Method	Bunun	Duoxu	Mboshi	Saisiyat	Seediq
Deleted	*Unfiltered*	**0.2573**	**0.4464**	0.4717	**0.2600**	**0.1899**
	Random	0.4620	0.5008	0.5121	0.3943	0.3075
	PDM	0.2754	0.4478	**0.4446**	0.2779	0.1947
	CTC	0.3025	0.4562	0.6048	0.3002	0.2325
Cropped	*Unfiltered*	0.3065	**0.4514**	0.4268	0.3700	0.2692
	Random	0.3372	0.4911	0.5402	0.3428	0.2473
	PDM	**0.3062**	0.5311	**0.4198**	**0.2908**	0.2049
	CTC	0.3351	0.4806	0.7046	0.3387	**0.2034**
Swapped	*Unfiltered*	0.4743	0.5759	0.5562	0.4500	0.3356
	Random	0.4584	0.6538	0.5513	0.3994	0.4881
	PDM	**0.2940**	**0.4691**	**0.4951**	**0.2106**	**0.2036**
	CTC	0.3014	0.5800	0.5396	0.3421	0.3207

Table 3: WER for each combination of simulated corruption setting and filtering method for each of the five CURATED datasets. This same information is visualized in bar graph format in Figure 2. The lowest WER in for each language/corruption is boldfaced.

Is That Your Final Answer? Test-Time Scaling Improves Selective Question Answering

William Jurayj **Jeffrey Cheng** **Benjamin Van Durme**
Johns Hopkins University
{wjurayj1,jcheng71,vandurme}@jhu.edu

Abstract

Scaling the test-time compute of large language models has demonstrated impressive performance on reasoning benchmarks. However, existing evaluations of test-time scaling make the strong assumption that a reasoning system should always give an answer to any question provided. This overlooks concerns about whether a model is *confident* in its answer, and whether it is appropriate to always provide a response. To address these concerns, we extract confidence scores during reasoning for thresholding model responses. We find that increasing compute budget at inference time not only helps models answer more questions correctly, but also increases confidence in correct responses. We then extend the current paradigm of *zero-risk* responses during evaluation by considering settings with non-zero levels of response risk, and suggest a recipe for reporting evaluations under these settings.[1]

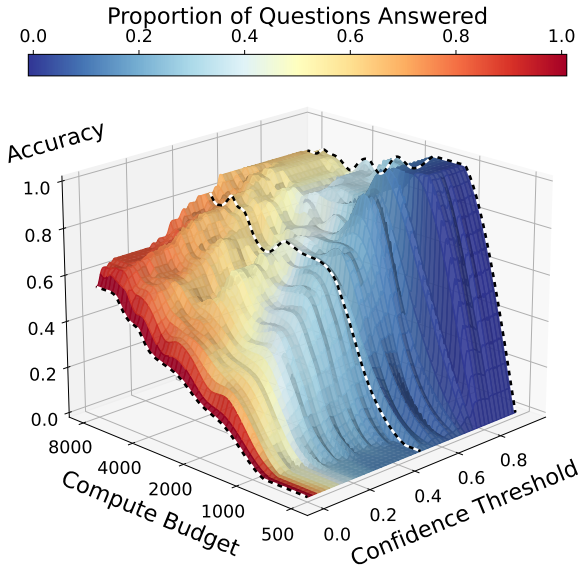


Figure 1: **DeepSeek R1-32B's accuracy is a function of compute budget and confidence threshold.** Increased confidence thresholds generally yield increased accuracy at the cost of response rate, while increased compute budgets sometimes decrease accuracy as they increase response rate. The vertical axis measures the accuracy of answered questions at a compute budget and confidence threshold. Color indicates the proportion of questions that are answered; in redder regions, the model is more likely to answer, whereas in bluer regions the model is less likely to answer.

1 Introduction

Scaling up language model inference-time compute using lengthy chains of thought has delivered impressive results on mathematical reasoning benchmarks that resisted training compute scaling (DeepSeek-AI et al., 2025; Muennighoff et al., 2025). These results, however, are reported in the zero-risk response setting: with no penalties for incorrect answers, the system always guesses even when it is not confident in its answer. In practice, this behavior is not always desirable.

Many question answering settings associate incorrect answers with measurable costs, ranging from low-risk responses found in game shows (Ferrucci et al., 2010) to high-stakes responses that can alter people's lives (Northpointe, 2017). *Selective question answering* addresses these challenges by allowing a model to refrain from answering questions which it might answer incorrectly (Kamath et al., 2020). This requires a selection function, which considers risk tolerance, coverage goals, and candidate answer confidence to decide whether a prediction should be given (Geifman and El-Yaniv, 2017). Knowing when not to answer is a critical quality for systems to collaborate effectively with humans (Verma et al., 2023), especially for test-time scaling systems that must constantly decide between refusing to answer and expending further compute to search for a possible solution.

To help address this issue, we evaluate test-time scaling models using a simple class of selection

[1]Code released at https://github.com/wjurayj/final_answer

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 636–644
July 27 - August 1, 2025

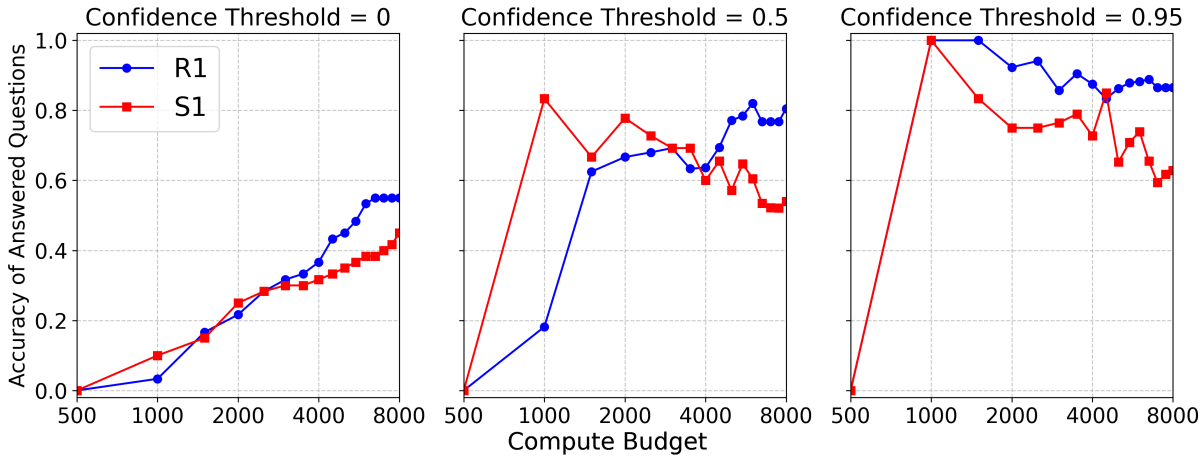


Figure 2: **Confidence thresholds on test-time scaling.** *(left)* When the confidence threshold is 0, the model answers 100% of questions. This is the only performance curve that is reported by test-time scaling research. *(center)* At a moderate threshold, more frequent absentions allow higher response accuracy. *(right)* At a high threshold, small amounts of test-time compute deliver very high accuracy at low answer rates, while test-time scaling provides more answers at the cost of answer accuracy.

functions that reject questions if a model is not confident in its answer after expending its compute budget. We evaluate these systems at different compute budgets, showing a new axis of model performance that answer accuracy alone struggles to measure. We suggest a class of utility functions that represent various levels of error risk to empirically measure the performance of these systems in settings where incorrect answers are penalized. Evaluation in these settings shows how compute scaling affects confidence in existing systems. Based on these insights, we propose a standard method for measuring model performance in settings with non-zero response risk. In summary we:

- Conduct the first evaluation of LLM test-time compute scaling on selective question answering, finding that increasing inference compute can help models distinguish between their correct and incorrect answers. (Section 3)
- Introduce evaluation settings that penalize incorrect answers and allow abstentions to help holistically evaluate models capable of scaling test-time compute. (Section 4)
- Invite the community to report test-time scaling performance on selective question answering under "Jeopardy Odds", which incentivize confidence calibration by penalizing incorrect answers while rewarding correct answers.

2 Methods

We explore how increasing compute budgets affects a model's performance on question answering tasks at different confidence thresholds. The choice of a budget and threshold is a *test-time* decision. We describe methods to quantify the two factors below:

Compute Budget refers to the amount of compute expended by the model at inference time. In all cases, we quantify a model's budget by counting the *number of tokens* in its reasoning trace. We use methods proposed by Muennighoff et al. (2025) to strictly enforce compute budgets. Specifically, we ignore any predicted end-of-thinking delimiters and instead append the token "Wait" if a model attempts to end its reasoning trace before reaching the budget, and we force decode the end-of-thinking delimiter once the budget is reached.

Confidence Threshold refers to the uncertainty of the model in its decoded answer. We quantify a model's confidence as the *sum of the log-probabilities* corresponding to the answer tokens.[2] For a confidence threshold, our selection function (Geifman and El-Yaniv, 2017) only accepts answers that the model delivers with confidence greater than its threshold, abstaining otherwise.

3 Experiments

3.1 Experimental Setup

We evaluate Deepseek-R1-32B (DeepSeek-AI et al., 2025) and s1 (Muennighoff et al., 2025) due to their exhibited test-time scaling capabilities and open-weight checkpoints, and choose

[2]Every answer in our dataset is a 3-digit number between 000 and 999, so consists of the same number of tokens.

Figure 3: **Test-time scaling improves confidence in correct answers on R1-32B (left) and S1-32B (right).** Each dot represents the model's confidence in an answer after spending a fixed amount of compute. Indigo series are correct answers, while orange series are incorrect. Dotted lines plot confidence trajectories for 10 randomly selected questions, emphasizing how confidence in correct versus incorrect answers changes with test-time compute. Note that individual answers may turn from orange to indigo if the model changes its prediction after thinking longer.

AIME 2024 and 2025 as our primary evaluation dataset. This dataset contains 60 hard math problems on which performance substantially benefits from larger compute budgets, making it a popular benchmark for evaluating test-time scaling. Additional experiments on GPQA (Rein et al., 2024) are included in Appendix B. We test the set of confidence thresholds $\{0.0, 0.5, 0.95\}$ across compute budgets within the range $[500, 8000]$, incrementing by 100 tokens. For a given budget and threshold, we report the accuracy of *answered* questions, treating never answering as yielding accuracy 0. As the number of answered questions differs across confidence thresholds, we note that accuracies are *not directly comparable* between models and compute budgets.

We use widely available open-source libraries to run our experiments, including HuggingFace Transformers (Wolf et al., 2020) and vLLM (Kwon et al., 2023) for language model inference, and the Language Model Evaluation Harness (Gao et al., 2024) to sample reasoning chains at temperature 0 and 32-bit precision. In particular, we use the variant of this library released by Muennighoff et al. (2025), and run a subset of the experiments that they run. We run experiments on 4 H100 GPUs.

3.2 Results

Figure 2 compares the accuracy of answers provided by R1-32B and S1-32B at different test-time compute budgets. When the confidence threshold is 0, models answer every question, so accuracy increases consistently with compute budget. We observe that these subplots are slices of a surface parameterized by compute budget and confidence threshold, shown in Figure 1. While higher confidence thresholds prevent the model from answering at low budgets, scaling compute at high thresholds delivers a larger volume of accurate answers. However, at higher confidence thresholds increased compute budget can actually decrease answer accuracy. This decrease in accuracy of yielded answers does not necessarily reflect decreased performance at higher budgets, but instead that the additional questions answered are less likely to be correct than those answered at lower budgets.

To investigate whether excessive thinking harms accuracy drops by pushing models to abandon correct answers, we plot how a model's confidence in individual answers moves over time. Figure 3 shows the answer confidences given by both models at varying compute budgets, colored according to their correctness, with a curve fit to the distribution. We note that as compute budget increases, the average confidence of its correct answers increases even as additional correct answers are discovered. Notably, this is not a universal property of test-time scaling models: S1-32B does not separate its correct answers from its incorrect answers as well as R1-32B.

4 Utility

4.1 Motivation

When refusal to answer is an option, accuracy can be trivially optimized by a system that answers extremely infrequently. Thus, a useful metric must capture both the accuracy of answers provided and

the system's propensity to provide answers. Many real world scenarios reward correct answers, but incur measurable costs for incorrect answers. We show our results involving confidence thresholds can be adapted to these settings.

Given a model $\mathcal{M}$ and an instance x of a task t, we define a *utility function* f to be

$$f(\mathcal{M}, x) = \begin{cases} 1 & \mathcal{M} \text{ answers } x \text{ correctly} \\ 0 & \mathcal{M} \text{ abstains from answering } x \\ r_t & \mathcal{M} \text{ answers } x \text{ incorrectly} \end{cases}$$

We can assume the reward for correct answers is 1 without loss of generality due to scaling. While there exist scenarios where refusing to answer also incurs a cost, this paper will only discuss the consequences when no extra cost is incurred; the conclusions we draw can be extended to these cases.

4.2 Problem Scenarios

We discuss three settings with varying risk levels:

- Exam Odds ($r_t = 0$): There are no costs incurred by incorrect answers. These are tasks where guessing isn't punished and the model should always try to provide a solution.
- Jeopardy Odds[3] ($r_t = -1$): The cost of an incorrect answer is equal to the reward for a correct answer. In these scenarios, no answer at all is preferable to an incorrect answer.
- High-Stakes Odds ($r_t = -20$): The cost of an incorrect answer far outweighs the reward for a correct answer. In this case, the model should answer only if absolutely certain.

4.3 Results

We keep the same experimental setup as described in Section 3.1. Rather than reporting accuracy, we instead report the utility in the three scenarios above, shown in Figure 4. We focus on the Jeopardy setting because it highlights why a system might choose not to answer; results in the other settings are in Appendix A.

The Exam setting's utility function does not distinguish refusal from incorrectness, so optimal performance is achieved trivially at confidence threshold to 0 so that every question gets the model's best guess. In the Jeopardy setting, however, this is non-trivial. We illustrate the complete function mapping compute budget and confidence threshold to Jeopardy performance in Figure 4: the checkered lines

[3]Inspired by the wagers made in the game show's 'Final Jeopardy' stage

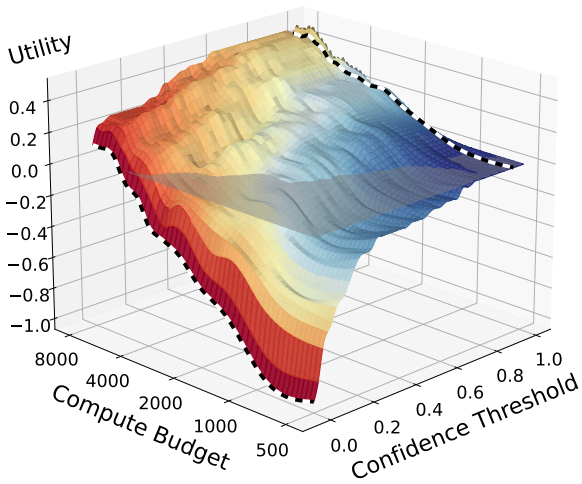


Figure 4: **Utility Surface of DeepSeek R1-32B for Jeopardy.** The vertical axis indicates performance in the Jeopardy setting at different compute budgets and confidence thresholds. The color indicates the proportion of questions that are answered, as in Figure 1. The horizontal plane divides positive and negative utility regions of the operating curve. The checkered lines show the confidence slices that we compare to s1 in Figure 5.

on this surface indicate the two slices that compose R1-32B's portion of Figure 5. We do not suggest that our choice of 0.95 is the optimal threshold for this task, or even that a threshold is the right approach to confidence calibration. Rather, we apply this naive method to show how test-time scaling for selective classification can benefit a practical question-answering setting.

We see on the left of Figure 2 that in the commonly reported Exam Odds, R1-32B and S1-32B scale comparably at threshold 0. In Jeopardy Odds, selective question answering at threshold 0.95 dramatically improves performance for both models. Additionally, although the two models scale comparably at Exam Odds, R1-32B substantially outperforms S1-32B at larger budgets in this new evaluation setting. Previous work overlooks this comparison. We call on future test-time compute scaling research to report optimal utility at Jeopardy Odds in addition to Exam Odds, to help readers understand performance across confidence demands.

5 Related Work

As scaling training compute has become prohibitively expensive (Hoffmann et al., 2022), models that scale performance with test-time compute have become a new frontier (Snell et al., 2024; Wu et al., 2025). These methods have delivered state-of-the-art results on hard reasoning tasks us-

ing lengthy chains of thought (DeepSeek-AI et al., 2025; Muennighoff et al., 2025). Current work in this space optimizes for question answering tasks which do not penalize incorrectness, ignoring settings that favor refusal over wrong answers (Ferrucci et al., 2010; Rajpurkar et al., 2018; Kamath et al., 2020). We draw motivation from methods for cost-sensitive learning (Mienye and Sun, 2021) and selective classification (Geifman and El-Yaniv, 2017), which navigate penalties for failure. These settings reward confidence calibration, which can be critical for effective collaboration with human experts (Verma et al., 2023). We are the first to investigate how serialized test-time compute helps models identify when they should not answer.

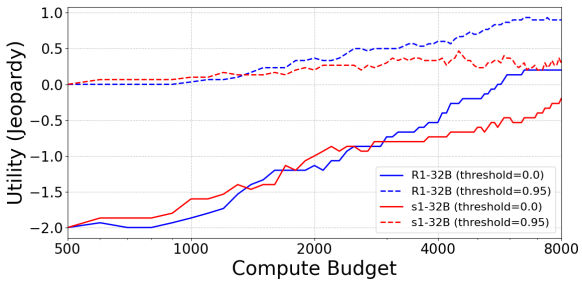


Figure 5: **Jeopardy utility scales differently across models and thresholds.** Performance of S1-32B and R1-32B in the Jeopardy odds setting under different confidence thresholds. Although S1 is competitive at lower budgets when the confidence threshold is 0, a higher threshold shows R1's superior scaling performance in a selective setting.

5.1 Test-time Scaling

Many methods for scaling test-time compute have been explored. These include searching over possible generations (Wang et al., 2024), sampling many completions and selecting the best answer among them (Wu et al., 2025), making gradient updates at inference time (Akyürek et al., 2024; Li et al., 2025), using reinforcement learning to incentivize generating chains of thought before answering (DeepSeek-AI et al., 2025), and simply fine-tuning on longer chains of thought (Muennighoff et al., 2025). Our work considers models fine-tuned on very long reasoning chains, and augments them with the ability to refuse to answer questions where they lack confidence. Concurrent work finds that such models can learn to predict when their answers are unlikely to be correct (Zhang et al., 2025; Huang et al., 2025), but do not show how this affects performance as additional compute is expended. In contrast, we show how model confidence scales with test-time compute, and demonstrate its value for optimizing performance in settings that allow refusal to answer.

5.2 Selective Question Answering

Refusing to answer is an important option in many prior works on question answering. SQuAD 2.0 included this feature by asking questions which have no answer, although they treat abstaining from answering as a correct answer in these unanswerable cases (Rajpurkar et al., 2018). Gameshow based research efforts use an approach more closely aligned with ours, which penalizes systems for answering incorrectly to encourage abstentions when a system cannot develop sufficiently high confidence (Ferrucci et al., 2010; Ferrucci, 2012; Boyd-Graber and Börschinger, 2020; Rodriguez et al., 2021). Related to quiz game settings is research into selective classification, which evaluates models performance across the coverage-accuracy curve, rather than at single point (Geifman and El-Yaniv, 2017). These approaches can be useful for avoiding costly errors in high-pressure domains (Khan et al., 2018), under distribution shift (Ren et al., 2023), or when designing systems that defer to expert humans when it lacks confidence that its input will be helpful (Mozannar and Sontag, 2020). Recent research in language modeling has investigated training language models to refuse to answer (Cao, 2024), and this capacity for refusal has become a point of competition among top industrial labs (Wei et al., 2024). However, this line of work does not investigate this behavior in sequential test-time scaling models on reasoning intensive tasks, where a model might find a confident answer given higher compute budgets.

6 Conclusion

We highlight a region of performance that is currently unexplored by test-time scaling research. We encourage the test-time scaling community to adopt these insights by reporting model scaling performance on benchmarks at both Exam Odds and Jeopardy Odds, to highlight their systems ability to scale confidence with test-time compute. Future work should focus on efficiently allocating test-time compute to meet confidence demands, and could investigate how test-time confidence scaling models should decide between extending reasoning and deferring to human experts.

Limitations

The selection function we implement is based entirely on the likelihood that a large language model assigns a series of tokens after thinking, which is not necessarily the optimal method for model confidence estimation. Furthermore, the Chain-of-Thought scaling method we apply may struggle to generalize to problem types that a model has note The method we use for 'budget forcing' (Muennighoff et al., 2025) may diminish performance by abruptly truncating chains of thought and driving the model outside of its training distribution. Concurrent work has introduced more elegant forms of compute budget control (Aggarwal and Welleck, 2025; Hou et al., 2025). Furthermore, we do not consider how compute costs might be incorporated in the model's utility function, which could encourage increased energy consumption. Finally, we recognize that by evaluating only on English questions and answers, we may miss model capabilities or weaknesses in lower-resource languages or in multilingual settings.

Acknowledgments

This work was funded in part by the U.S. National Science Foundation under grant No. 2204926, and by the Defense Advanced Research Project Agency (DARPA) SciFy program (contract No. HR001125C0304) and CODORD program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or DARPA. We thank Miriam Wanner, Zhengping Jiang, Orion Weller, Marc Marone, Alex Martin, and Beepul Bharti for helpful conversations throughout this project.

References

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling How Long A Reasoning Model Thinks With Reinforcement Learning. *arXiv preprint*. ArXiv:2503.04697 [cs].

Ekin Akyürek, Mehul Damani, Linlu Qiu, Han Guo, Yoon Kim, and Jacob Andreas. 2024. The Surprising Effectiveness of Test-Time Training for Abstract Reasoning. *arXiv preprint*. ArXiv:2411.07279 [cs].

Jordan Boyd-Graber and Benjamin Börschinger. 2020. What question answering can learn from trivia nerds. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7422–7435, Online. Association for Computational Linguistics.

Lang Cao. 2024. Learn to Refuse: Making Large Language Models More Controllable and Reliable through Knowledge Scope Limitation and Refusal Mechanism. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3628–3646, Miami, Florida, USA. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint*. ArXiv:2501.12948 [cs].

D. A. Ferrucci. 2012. Introduction to "this is watson". *IBM Journal of Research and Development*, 56(3.4):1:1–1:15.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. 2010. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. A framework for few-shot language model evaluation.

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4885–4894, Red Hook, NY, USA. Curran Associates Inc.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. 2025. ThinkPrune: Pruning Long Chain-of-Thought of LLMs via Reinforcement Learning. *arXiv preprint*. ArXiv:2504.01296 [cs].

Chengsong Huang, Langlin Huang, Jixuan Leng, Jiacheng Liu, and Jiaxin Huang. 2025. Efficient test-time scaling via self-calibration. *Preprint*, arXiv:2503.00031.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

Salman H. Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A. Sohel, and Roberto Togneri. 2018. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8):3573–3587.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Wen-Ding Li, Keya Hu, Carter Larsen, Yuqing Wu, Simon Alford, Caleb Woo, Spencer M. Dunn, Hao Tang, Wei-Long Zheng, Yewen Pu, and Kevin Ellis. 2025. Combining induction and transduction for abstract reasoning. In *The Thirteenth International Conference on Learning Representations*.

Ibomoiye Domor Mienye and Yanxia Sun. 2021. Performance analysis of cost-sensitive learning methods with application to imbalanced medical data. *Informatics in Medicine Unlocked*, 25:100690.

Hussein Mozannar and David Sontag. 2020. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint*. ArXiv:2501.19393 [cs].

Northpointe. 2017. Practitioner's guide to compas core.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. 2023. Out-of-distribution detection and selective generation for conditional language models. In *The Eleventh International Conference on Learning Representations*.

Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan Boyd-Graber. 2021. Quizbowl: The Case for Incremental Question Answering. *arXiv preprint*. ArXiv:1904.04792 [cs].

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Rajeev Verma, Daniel Barrejón, and Eric Nalisnick. 2023. Learning to Defer to Multiple Experts: Consistent Surrogate Losses, Confidence Calibration, and Conformal Ensembles. *arXiv preprint*. ArXiv:2210.16955 [stat].

Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah Goodman. 2024. Hypothesis search: Inductive reasoning with language models. In *The Twelfth International Conference on Learning Representations*.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *Preprint*, arXiv:2411.04368.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2025. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *The Thirteenth International Conference on Learning Representations*.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025. Reasoning models know when they're right: Probing hidden states for self-verification. *Preprint*, arXiv:2504.05419.

A Appendix

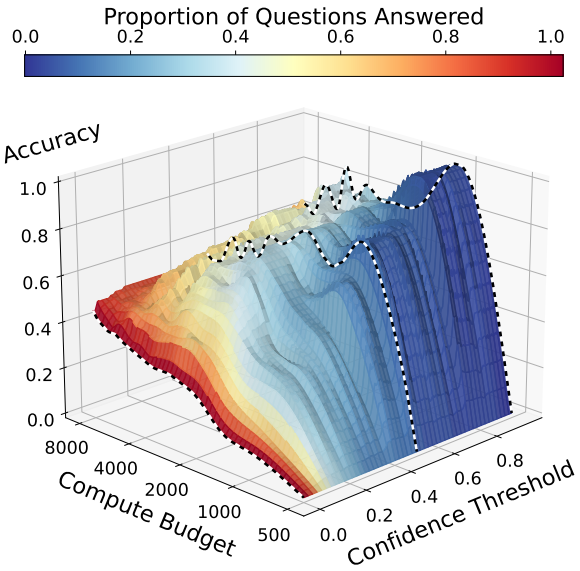
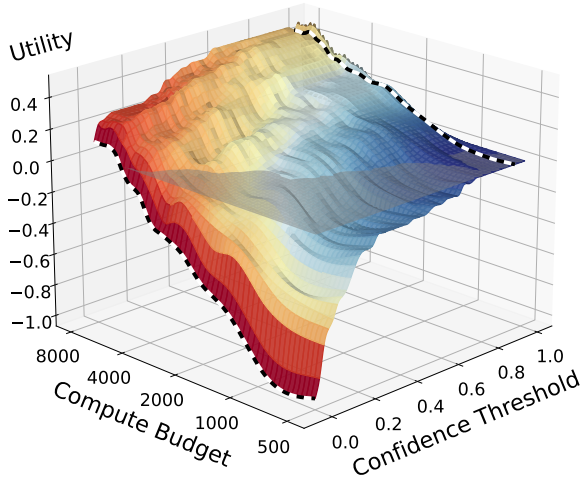


Figure 6: **(above) S1-32B's answer accuracy is a function of compute budget and confidence threshold.** This plot corresponds to the R1-32B plot in Figure 1. **(below) Utility surface of S1-32B for Jeopardy.** This plot corresponds to the R1-32B plot in Figure 4.



B GPQA Experiments

GPQA consists of 448 graduate level multiple choice questions in the fields of physics, biology, and chemistry. These questions are considered "Google-proof", in that skilled non-experts with access to the open internet struggle to answer them. We run experiments on the 'Diamond' subset of this dataset, which consists of the 198 questions which had the clearest answers to domain experts, while being the most difficult for non-experts to answer; this subset has also served as a benchmark for test-time scaling language models (Muennighoff et al., 2025). Our experiments on GPQA follow

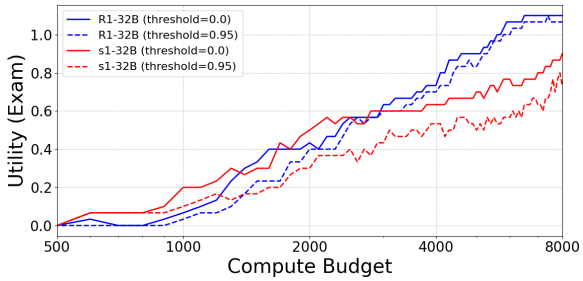
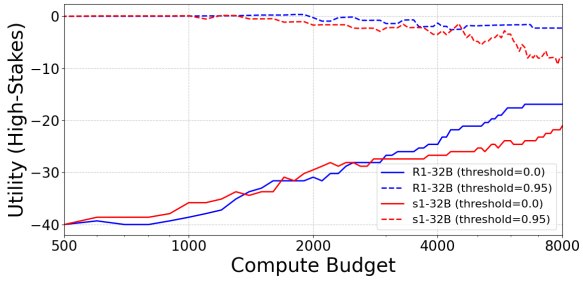


Figure 7: **Additional Model Comparisons.** We additionally compare performance of S1-32B and R1-32B in the Exam Odds *(above)* and High-Stakes Odds *(below)* settings under different confidence thresholds. Like Jeopardy odds depicted in Figure 5, High-Stakes Odds illustrates a performance distinction at high confidence thresholds that is not evident from conventional Exam odds.



the same basic procedure described in Section 3, except that we stop evaluate token budgets in range [500, 4000].

As with AIME, Figure 8 shows how models can increase performance by only answering when highly confident. Notably, there is very little gap between thresholds 0 and 0.5 on GPQA, owing to its multiple choice format. Moreover, whereas R1-32B outperformed S1-32B at all confidence thresholds, R1-32B only exceeds S1-32B at a confidence threshold of 0.95.

Figure 9 shows performance at Jeopardy odds of R1-32B (above) and S1-32B (below). Although both models perform similarly at low thresholds, R1-32B's superior test-time scaling behavior becomes evident at higher confidence thresholds, allowing it to achieve higher peak utilities compared to S1-32B. This gap is reflected in the slices shown in Figure 10. We also note that the slices of the surface are nearly identical at lower confidence thresholds below 0.25. This is likely due to the multiple choice format of GPQA; when the model does not know the answer to a question, it assigns roughly equal probability to the four possible answers.

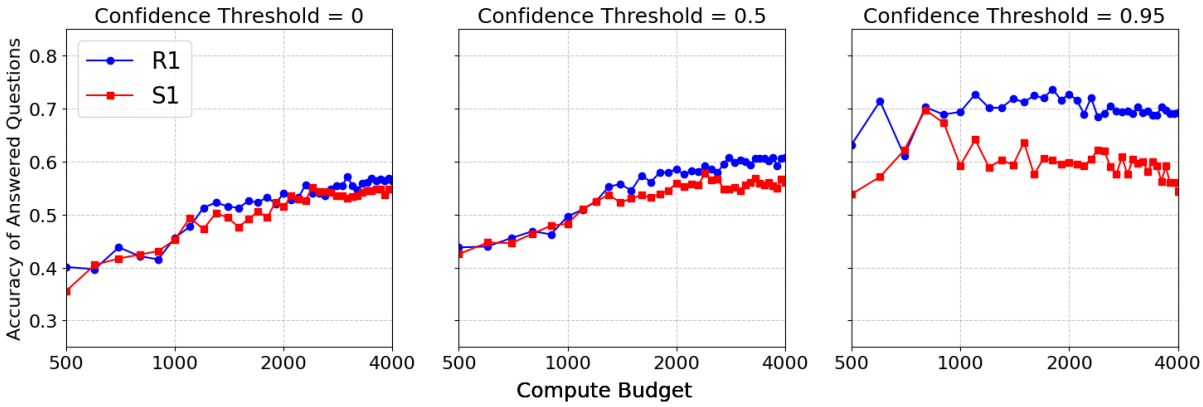


Figure 8: **Confidence thresholds on test-time scaling.** *(left)* When the confidence threshold is 0, the model answers 100% of questions. This is the only performance curve that is reported by test-time scaling research. *(center)* At a moderate threshold, more frequent absentions allow higher response accuracy. *(right)* At a high threshold, small amounts of test-time compute deliver very high accuracy, while test-time scaling provides more answers at the cost of answer accuracy. We treat the decision to never answer as yielding accuracy 0.

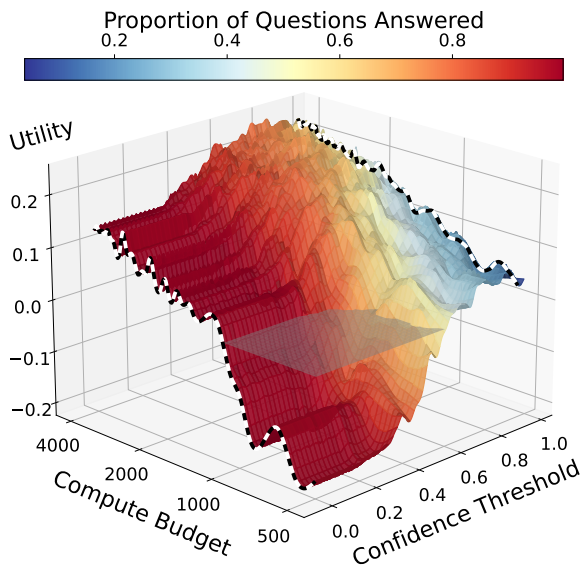
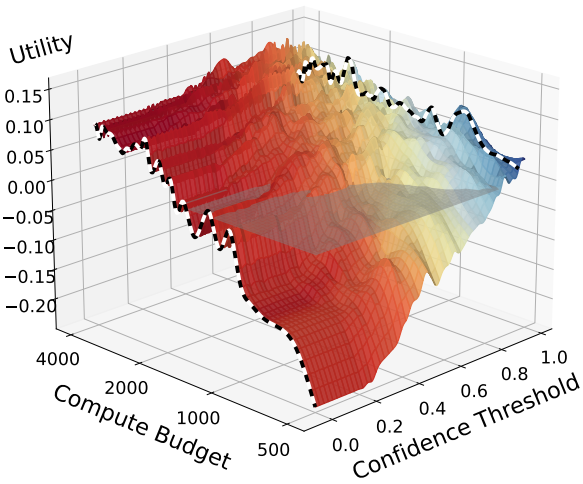


Figure 9: **Utility surfaces of R1-32B (above) and S1-32B (below) for Jeopardy utility on GPQA**. Utility is a function of compute budget and confidence threshold. These plots mirror the surfaces in Figure 4 and Figure 6.



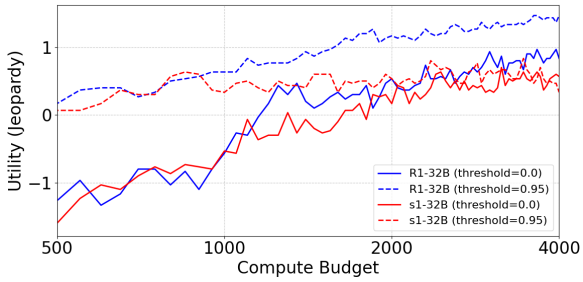


Figure 10: **Jeopardy utility on GPQA scales differently across models and thresholds.** Performance of S1-32B and R1-32B on GPQA in the Jeopardy odds setting under different confidence thresholds. While S1 is competitive in the case when threshold is 0, a higher threshold shows R1's superior scaling performance.

Acoustic Individual Identification of White-Faced Capuchin Monkeys Using Joint Multi-Species Embeddings

Álvaro Vega-Hidalgo[1], Artem Abzaliev[1], Thore Bergman[2,3], Rada Mihalcea[1]
[1]Computer Science and Engineering, University of Michigan
[2]Department of Psychology, University of Michigan
[3]Department of Ecology and Evolutionary Biology, University of Michigan
{alvarovh, abzaliev, thore, mihalcea}@umich.edu

Abstract

Acoustic individual identification of wild animals is an essential task for understanding animal vocalizations within their social contexts, and for facilitating conservation and wildlife monitoring efforts. However, most of the work in this space relies on human efforts, as the development of methods for automatic individual identification is hindered by the lack of data. In this paper, we explore cross-species pre-training to address the task of individual classification in white-faced capuchin monkeys. Using acoustic embeddings from birds and humans, we find that they can be effectively used to identify the calls from individual monkeys. Moreover, we find that joint multi-species representations can lead to further improvements over the use of one representation at a time. Our work demonstrates the potential of cross-species data transfer and multi-species representations, as strategies to address tasks on species with very limited data.

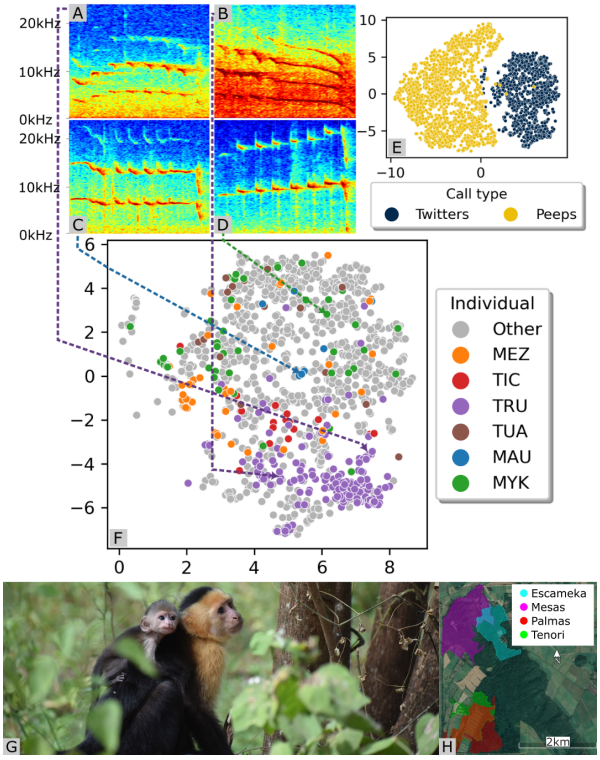


Figure 1: (A–D) Capuchin Twitter vocalizations show diverse structural variations. (E–F) t-SNE of Google Perch-Whisper embeddings. (E) Call type clusters. (F) Colored by individual, highlighting four diverse examples of Twitters. (G) Adult female capuchin with infant in Taboga. (H) Territories of four capuchin groups in the Taboga Reserve in northwestern Costa Rica.

1 Introduction

For a long time, researchers viewed the vocalizations of non-human species as mere reactions to internal emotional states (Lorenz, 1952). Consequently, early scientific methods in animal communication research largely overlooked individual differences and did not test for the presence of linguistic features (e.g., pragmatics, semantics, syntax) in animal communication systems. This simplified view of animal communication has been overturned by the growing evidence uncovering the presence of linguistic features in non-human animals (Bergman et al., 2019), leading to the emergence of Animal Linguistics as a formal interdisciplinary research field (Bowling and Fitch, 2015; Engesser et al., 2015; Suzuki, 2024; Berthet et al., 2023; Suzuki, 2021; Scott-Phillips and Heintz, 2023). This shift in perspective highlights the need for individual-level analysis, as it allows researchers to account for the social and environmental contexts in which vocalizations occur, ultimately improving our ability to test their linguistic capacities more rigorously.

Additionally, long-term, individual-level analyses are critical for understanding and protecting wildlife. Such analyses support key approaches like social network quantification, assessing animal cognition, and performing capture–recapture techniques for tracking population dynamics (Slater, 1981; Carlson et al., 2020). Over the past decade,

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 645–659
July 27 - August 1, 2025

acoustic monitoring has emerged as a widely adopted, cost-efficient strategy in conservation, leading to growing interest in acoustic individual identification. By enabling researchers to recognize individuals from their vocalizations, this approach paves the way for more nuanced insights into ecology, behavior, evolution, and conservation (Knight et al., 2024).

In this paper, we address the task of acoustic individual identification in white-faced capuchins. We collect a two-year dataset of individualized focal recordings, a labor-intensive yet optimized method that mitigates signal-to-noise and cocktail party problems in wild bioacoustic settings (Bermant, 2021; Miron et al., 2024). Using this dataset, we evaluate human speech- and bird bioacoustic-based pre-trained networks, comparing single-embedding models to ensembles that merge embeddings from distinct networks. We hypothesize that human speech embeddings, such as Whisper or HuBERT, complement bioacoustic embeddings like Google Perch or BirdNET–originally trained on bird sounds–and predict that heterogeneous embedding combinations will outperform single-embedding models.

Transfer learning has significantly advanced acoustic classification tasks in non-human animals (Miyaguchi et al., 2024; Kahl et al., 2023; Abzaliev et al., 2024). Recent studies on gibbons have explored the use of self-supervised speech models (e.g., HuBERT, Wav2vec 2.0), pre-trained bird classifiers (e.g., BirdNET, Perch), and non-transfer-learning deep models for primate acoustic identification, finding that speech models most effectively capture individual vocal signatures, bird classifiers perform well in automated detection but are more susceptible to background noise, and non-transfer-learning models struggle when trained on small datasets (Cauzinille et al., 2024; Clink et al., 2024). Nevertheless, it remains unclear whether using multiple joint embeddings leads to better performance by exploiting complementary features from different training data domains.

This work makes three main contributions. First, we propose white-faced capuchin monkeys as a model organism for advancing computational research on animal communication. Second, we show that combining embeddings from human speech and bird bioacoustics models significantly improves acoustic identification performance in white-faced capuchins, outperforming single-embedding baselines. Finally, our findings show that acoustic diversity and soundscape similarity play a greater role than phylogenetic proximity. Smaller models trained on diverse bird vocalizations recorded in natural environments outperform much larger speech-trained models designed for humans, despite humans being more closely related to our study species. These results highlight the value of cross-species model development in achieving better generalization for the acoustic identification task.

2 Study system: white-faced capuchin monkeys in the Taboga Reserve, Costa Rica

White-faced capuchin monkeys (*Cebus capucinus*) are ideal for studying animal communication, with 27 call types (Gros-Louis et al., 2008), complex social behavior and cognition including tool use (Goldsborough et al., 2024), complex social networks (Crofoot et al., 2011) and cultural transmission (Perry et al., 2017). Taboga hosts their highest known density (Tinsley Johnson et al., 2020).

Data collection. Our field team collected audio recordings of focal individuals by following them in the Taboga forest. We used directional microphones aimed at the subjects from January 2021 to December 2022 through the wet and dry seasons, with hours ranging from 5 am to 5 pm. Recordings were captured at $48\,\mathrm{kHz}$ and $16\,\mathrm{bit}$ resolution. These raw recordings were subsequently trimmed to isolate the precise moments when vocalizations were detected, and only the calls classified as either a "Peep" or "Twitter" were included in this dataset, according to established criteria in the literature (Gros-Louis et al., 2008).

Audio recordings. The full dataset consists of 1,257 Twitter recordings and 2,089 Peep recordings from 45 individuals, although 15% of the recordings were assigned to unknown individuals. We include data from individuals that had at least 30 recorded calls, while recordings from unidentified subjects encountered in the field are grouped into an "Unknown" class. For Peeps, this dataset includes 16 individuals, and for Twitters this dataset included 10 individuals (total sample=1609). Peep calls are typically short (mean $0.27\,\mathrm{s}$, SD $0.27\,\mathrm{s}$), whereas Twitter calls are more complex (Figure 1) and longer (mean $0.40\,\mathrm{s}$, SD $0.18\,\mathrm{s}$).

3 Cross-Species Embeddings for Individual Classification

Collecting focal audio recordings of wild animals in their natural habitat is a challenging and resource-intensive task. Even with dedicated field teams, building large enough datasets to fully exploit deep neural networks is difficult. As a result, transfer learning–which leverages the inductive bias of models pre-trained on larger, related datasets–has emerged as the most effective strategy for achieving high performance in bioacoustic classification under low-data conditions (Ghani et al., 2023a).

Audio Representation Models. We extract pre-trained embeddings from Google Perch V8 (Ghani et al., 2023a), a model primarily trained on bird vocalizations, and Whisper (Radford et al., 2022), which was predominantly pre-trained on human speech. While additional embeddings were evaluated, we focus on these two in the main text for clarity, with results from five other models detailed in Appendix A. We apply mean-pooling to obtain lower-dimensional representations from large speech models like Whisper.

Minimum Redundancy Maximum Relevance. To combine representations from multiple species, we explore a feature-select model using Minimum Redundancy and Maximum Relevance (MRMR) (Ding and Peng, 2005), alongside simple concatenation and summation. Originally developed in cancer research for gene selection, MRMR improves feature selection in high-dimensional datasets by balancing two key criteria: maximizing relevance to the target variable (measured via mutual information) while minimizing redundancy (filtered using a correlation coefficient threshold). Our implementation starts with the feature that has the highest mutual information among both embeddings, removes any features with a correlation coefficient of 0.8 or higher, and then iteratively selects the next most informative feature. This process continues until 1024 embedding features are selected from both embeddings, ensuring an optimal balance of diversity and informativeness.

Experimental Setup. To ensure a fair comparison, we carefully control parameter counts and apply hyperparameter tuning. Single-embedding models and the MRMR model compress each input into 512 units, then reduced it to 64 for final classification. Concatenation and summation ensembles apply a 256-dimensional compression to each embedding separately, then sum or concatenate the outputs before another 64-unit layer. For a robust comparison, we generate 50 random train-test splits (10 recordings per individual in the test set) and train models with all seven single embeddings as well as all pairwise combinations (concatenation, summation, and MRMR). To identify the best hyperparameters for each model trained, we conduct a search over learning rates $\{1\text{e-}5, 5\text{e-}5, 1\text{e-}4, 5\text{e-}4, 1\text{e-}3, 5\text{e-}3\}$ and dropout rates $\{0.2, 0.3, 0.4, 0.5, 0.6\}$, evaluating 30 randomly sampled configurations for 100 epochs each with early stopping (patience=10, min ΔF1=0.001), and selected the highest F1-scoring setup. All models are trained using the Adam optimizer. After confirming normality and homoscedasticity, we compare each architecture's top-performing model via ANOVA and a post-hoc Tukey test.

Whisper Layer Probing. To pinpoint which Whisper transformer layer encodes the richest individual-specific information, we trained Perch–Whisper MRMR models in which the Whisper input is systematically replaced with the hidden representation from each of the 33 encoder layers. For each layer, we retrain the model across the 50 random train-test splits using the same training schedule described above.

Spectrogram annotations and measurements. To compare explainable acoustic features with non-interpretable deep embeddings, we manually measure Peak Frequency and other acoustic parameters from spectrograms, following standard bioacoustic methods. Using Raven Pro 1.6 (K. Lisa Yang Center for Conservation Bioacoustics at the Cornell Lab of Ornithology, 2024), we select regions of interest and extract 30 interpretable features (see Appendix A), including Peak Frequency, Center Frequency, and Center Time. These measurements were taken from six individuals—one adult male, one adult female, and one infant from each of the two monkey troops—chosen for their distinct characteristics.

4 Results

Table 1 shows the results of the acoustic identification task for selected models. We present F1 scores for the models trained on bird vocalizations and

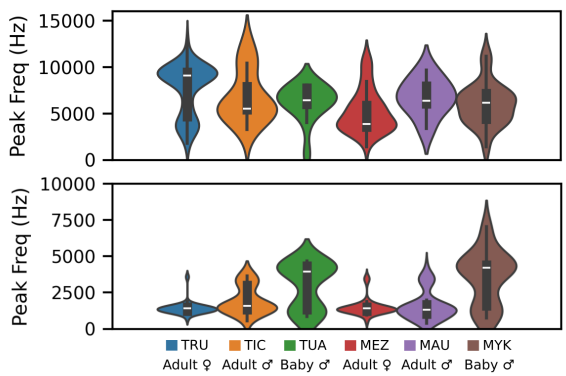


Figure 2: Peak frequency distributions for six capuchin monkey individuals, shown for Twitters (top) and Peeps (bottom) call types.

human speech data, together with their ensembles. While single-species vocalization models perform reasonably well, the models with the highest F1 scores are those that combine multiple embeddings (either using summation, concatenation or MRMR). Furthermore, the best-performing ensemble combine models developed for bioacoustic vocalizations and models developed for human speech. This highlights the potential of cross-species pre-training in a limited data regime. Pre-training on human speech does not capture enough information for the bioacoustic domain, as shown by the performance of Whisper for both vocalization types. But combined, those two models achieve an F1 score of 0.70 for Peeps and 0.66 for Twitters. This improved performance suggests that combining speech-trained and bioacoustic-trained embeddings effectively leverages complementary information. We also present the results for other models in Appendix A.

Despite its smaller size and more limited training dataset, the bioacoustic model Perch outperforms the much larger Whisper model, which was developed for human speech. Domain relevance is more important than model size, training data set size, or phylogenetic proximity for the acoustic identification task in Capuchins. Trained on data from noisy field conditions, Perch learns the acoustic variability of field conditions, contributing to its strong performance. Although our focal species is neither a bird nor a human, the top-performing models across architectures are trained using both bird- and human-derived embeddings, suggesting that joint multi-species embeddings provide better generalization for Capuchin acoustic classification tasks.

To better understand Whisper's contribution to these multi-species embeddings, we conducted a layer-wise probing analysis across 50 training runs. We found that intermediate layers yielded slightly better classification performance for both Peeps and Twitters (Figure 3), though differences across layers were relatively modest.

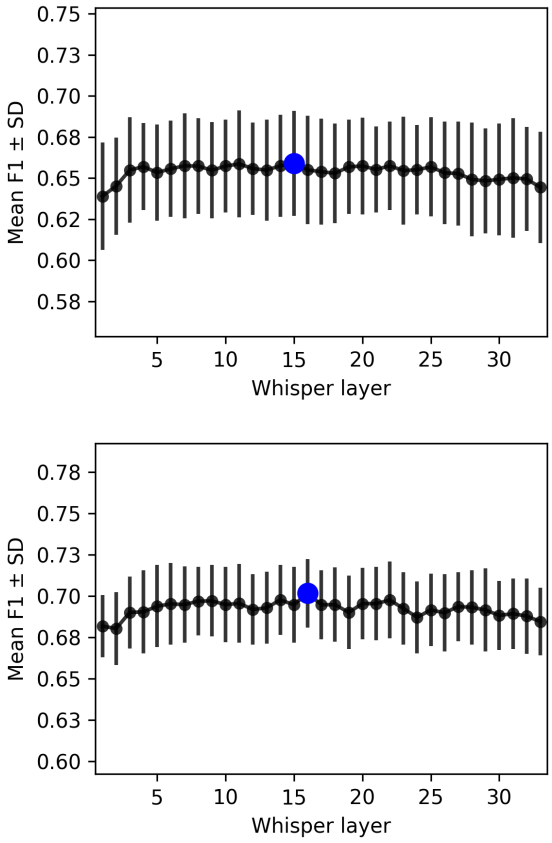


Figure 3: Whisper layer-wise probing (mean F1 across 50 random train-test splits) for Twitters (top) and Peeps (bottom). Intermediate layers yield the highest performance for individual classification (maximum value highlighted).

We visualize the embeddings of the best-performing model from table 1 using t-SNE (van der Maaten, 2009) in Figure 1. Different call types formed well-defined clusters (Figure 1E), whereas individual classifications appear more diffuse (Figure 1F), illustrating the difficulty of the acoustic identification task (see Appendix A for more t-SNE visualizations). We also analyze the distribution of peak frequencies across individuals in Figure 2. Lower-pitched sounds characterize Peeps, while Twitters span a broader spectral range of peak frequencies. Notably, both call types exhibit bimodal distributions, with this pattern being more pronounced in certain individuals. This bi-

Table 1: Top-performing models for Twitters and Peeps (Mean F1 Score ± SD), with significance assessed by comparison to the best simple model (Perch). Significance levels: * for $p < 0.05$ and ** for $p < 0.0001$ (Tukey's test).

Model	F1 Score
Twitters	
Chance (uniform 1/11)	0.09 ± 0.00
Perch (Simple)	0.61 ± 0.03
Whisper (Simple)	0.55 ± 0.03
Perch + Whisper (Concat)	0.63 ± 0.03
Perch + Whisper (Sum)	0.63 ± 0.03*
Perch + Whisper (MRMR)	**0.66 ± 0.03****
Peeps	
Chance (uniform 1/17)	0.06 ± 0.00
Perch (Simple)	0.66 ± 0.02
Whisper (Simple)	0.62 ± 0.03
Perch + Whisper (Concat)	0.67 ± 0.02*
Perch + Whisper (Sum)	0.68 ± 0.02**
Perch + Whisper (MRMR)	**0.70 ± 0.02****

modal distribution could reflect two or more call subtypes with distinct pitches and should be investigated further to test for the existence of pragmatics or semantics in their communication system through pitch modulation. Variability within the Twitter call type extends beyond overall pitch modulation. Some Twitters exhibit an n-shaped pitch contour, a continuous descending note, a final lower-pitched note, or a rising pitch throughout the call (Figure 1-A, B, C, D, respectively). Empirical studies incorporating rich social and environmental contexts will be crucial for uncovering the functional significance of this variation in Capuchin calls.

5 Conclusion

This study examined acoustic individual identification in two call types of white-faced capuchins. We established performance baselines for pre-trained embeddings and found that combining multiple embeddings (summation, concatenation, and minimum redundancy maximum relevance) improves classification performance. Our findings also indicate that domain relevance outweighs model size in noisy environments. Future work should extend these multi-species embeddings to other taxa, confirming broader applicability in bioacoustics and animal linguistics.

6 Limitations

While this study focused on acoustic identification, a deeper investigation into the behavioral and social functions of these call types remains relevant for future work. While there are other ways of improving acoustic identification, such as data augmentation (MacIsaac et al., 2024), we considered those techniques out of scope for the present study and focused on investigating the complementarity of joint multi-species embeddings. Our primary goal with this dataset is to make it accessible to the broader scientific community. We anticipate making it publicly available in a forthcoming study with further analyses.

7 Ethical Considerations

No animals were harmed during this study. All research adhered to ethical guidelines for animal welfare, recognizing the importance of studying animal communication while prioritizing their well-being, particularly in the context of climate change and habitat loss affecting this species. Additionally, all individuals involved in data collection and processing were engaged in formal employment or academic research under ethical labor practices.

Acknowledgments

We thank the entire Capuchinos de Taboga project team for their support and dedication throughout the data collection process. Their field expertise and commitment were essential to building the dataset that underpins this study. We also thank Robin Laurie for providing the capuchin monkey photograph used in Figure 1.

References

Artem Abzaliev, Humberto Pérez Espinosa, and Rada Mihalcea. 2024. Towards dog bark decoding: Leveraging human speech processing for automated bark classification. *arXiv preprint arXiv:2404.18739*.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Preprint*, arXiv:2006.11477.

Thore J Bergman, Jacinta C Beehner, Melissa C Painter, and Morgan L Gustison. 2019. The speech-like properties of nonhuman primate vocalizations. *Animal Behaviour*, 151:229–237.

Peter C Bermant. 2021. Biocppnet: automatic bioacoustic source separation with deep neural networks. *Scientific Reports*, 11(1):23502.

Mélissa Berthet, Camille Coye, Guillaume Dezecache, and Jeremy Kuhn. 2023. Animal linguistics: a primer. *Biological reviews*, 98(1):81–98.

Daniel L Bowling and W Tecumseh Fitch. 2015. Do animal communication systems have phonemes? *Trends in Cognitive Sciences*, 19(10):555–557.

Nora V Carlson, E McKenna Kelly, and Iain Couzin. 2020. Individual vocal recognition across taxa: a review of the literature and a look into the future. *Philosophical Transactions of the Royal Society B*, 375(1802):20190479.

Jules Cauzinille, Benoît Favre, Ricard Marxer, Dena Clink, Abdul Hamid Ahmad, and Arnaud Rey. 2024. Investigating self-supervised speech models' ability to classify animal vocalizations: The case of gibbon's vocal signatures. In *Interspeech 2024*, pages 132–136. ISCA; ISCA.

Dena J Clink, Hope Cross-Jaya, Jinsung Kim, Abdul Hamid Ahmad, Moeurk Hong, Roeun Sala, Hélène Birot, Cain Agger, Thinh Tien Vu, Hoa Nguyen Thi, et al. 2024. Benchmarking automated detection and classification approaches for monitoring of endangered species: a case study on gibbons from cambodia. *bioRxiv*, pages 2024–08.

Margaret C Crofoot, Daniel I Rubenstein, Arun S Maiya, and Tanya Y Berger-Wolf. 2011. Aggression, grooming and group-level cooperation in white-faced capuchins (cebus capucinus): Insights from social networks. *American Journal of Primatology*, 73(8):821–833.

Chris Ding and Hanchuan Peng. 2005. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205.

Sabrina Engesser, Jodie MS Crane, James L Savage, Andrew F Russell, and Simon W Townsend. 2015. Experimental evidence for phonemic contrasts in a nonhuman vocal system. *PLoS biology*, 13(6):e1002171.

Burooj Ghani, Tom Denton, Stefan Kahl, and Holger Klinck. 2023a. Global birdsong embeddings enable superior transfer learning for bioacoustic classification. *Scientific Reports*, 13(1):22876.

Burooj Ghani, Tom Denton, Stefan Kahl, and Holger Klinck. 2023b. Global birdsong embeddings enable superior transfer learning for bioacoustic classification. *Scientific Reports*, 13(1):22876.

Zoë Goldsborough, Margaret C Crofoot, and Brendan J Barrett. 2024. Male-biased stone tool use by wild white-faced capuchins (cebus capucinus imitator). *American Journal of Primatology*, 86(4):e23594.

Julie J Gros-Louis, Susan E Perry, Claudia Fichtel, Eva Wikberg, Hannah Gilkenson, Susan Wofsy, and Alex Fuentes. 2008. Vocal repertoire of cebus capucinus: acoustic structure, context, and usage. *International Journal of Primatology*, 29:641–670.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *Preprint*, arXiv:2106.07447.

Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. 2023. Masked autoencoders that listen. *Preprint*, arXiv:2207.06405.

K. Lisa Yang Center for Conservation Bioacoustics at the Cornell Lab of Ornithology. 2024. Raven pro: Interactive sound analysis software (version 1.6.5). [Computer software].

Stefan Kahl, Tom Denton, Holger Klinck, Hendrik Reers, Francis Cherutich, Hervé Glotin, Hervé Goëau, Willem-Pier Vellinga, Robert Planqué, and Alexis Joly. 2023. Overview of birdclef 2023: Automated bird species identification in eastern africa. In *CLEF (Working Notes)*, pages 1934–1942.

Stefan Kahl, Connor M. Wood, Maximilian Eibl, and Holger Klinck. 2021. Birdnet: A deep learning solution for avian diversity monitoring. *Ecological Informatics*, 61:101236.

Elly Knight, Tessa Rhinehart, Devin R de Zwaan, Matthew J Weldy, Mark Cartwright, Scott H Hawley, Jeffery L Larkin, Damon Lesmeister, Erin Bayne, and Justin Kitzes. 2024. Individual identification in acoustic recordings. *Trends in Ecology & Evolution*.

Konrad Lorenz. 1952. *King Solomon's Ring*. T. Y. Crowell, New York.

Jennifer MacIsaac, Stuart Newson, Adham Ashton-Butt, Huma Pearce, and Ben Milner. 2024. Improving acoustic species identification using data augmentation within a deep learning framework. *Ecological Informatics*, 83:102851.

Marius Miron, Sara Keen, Jen-Yu Liu, Benjamin Hoffman, Masato Hagiwara, Olivier Pietquin, Felix Effenberger, and Maddie Cusimano. 2024. Biodenoising: animal vocalization denoising without access to clean data. *arXiv preprint arXiv:2410.03427*.

Anthony Miyaguchi, Adrian Cheung, Murilo Gustineli, and Ashley Kim. 2024. Transfer learning with pseudo multi-label birdcall classification for ds@ gt birdclef 2024. *arXiv preprint arXiv:2407.06291*.

Susan E Perry, Brendan J Barrett, and Irene Godoy. 2017. Older, sociable capuchins (cebus capucinus) invent more social behaviors, but younger monkeys innovate more in other contexts. *Proceedings of the National Academy of Sciences*, 114(30):7806–7813.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *Preprint*, arXiv:2212.04356.

Thom Scott-Phillips and Christophe Heintz. 2023. Animal communication in linguistic and cognitive perspective. *Annual Review of Linguistics*, 9(1):93–111.

PJB Slater. 1981. Individual differences in animal behavior. In *Perspectives in Ethology: Volume 4 Advantages of Diversity*, pages 35–49. Springer.

Toshitaka N Suzuki. 2021. Animal linguistics: exploring referentiality and compositionality in bird calls. *Ecological Research*, 36(2):221–231.

Toshitaka N Suzuki. 2024. Animal linguistics. *Annual Review of Ecology, Evolution, and Systematics*, 55.

Elizabeth Tinsley Johnson, Marcela E Benítez, Alexander Fuentes, Celia R McLean, Ariek B Norford, Juan Carlos Ordoñez, Jacinta C Beehner, and Thore J Bergman. 2020. High density of white-faced capuchins (cebus capucinus) and habitat quality in the taboga forest of costa rica. *American Journal of Primatology*, 82(2):e23096.

Laurens van der Maaten. 2009. Learning a parametric embedding by preserving local structure. In *International Conference on Artificial Intelligence and Statistics*.

A Appendix

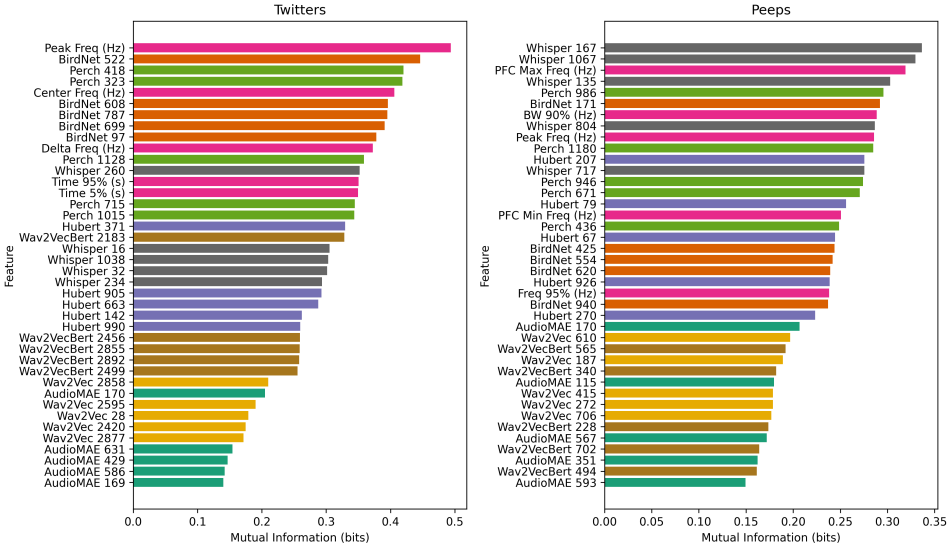


Figure 4: Mutual information of the top features for both call type datasets, spanning seven acoustic pre-trained embeddings. We display the five highest-performing features per pre-trained embedding, along with the top five interpretable features per model.

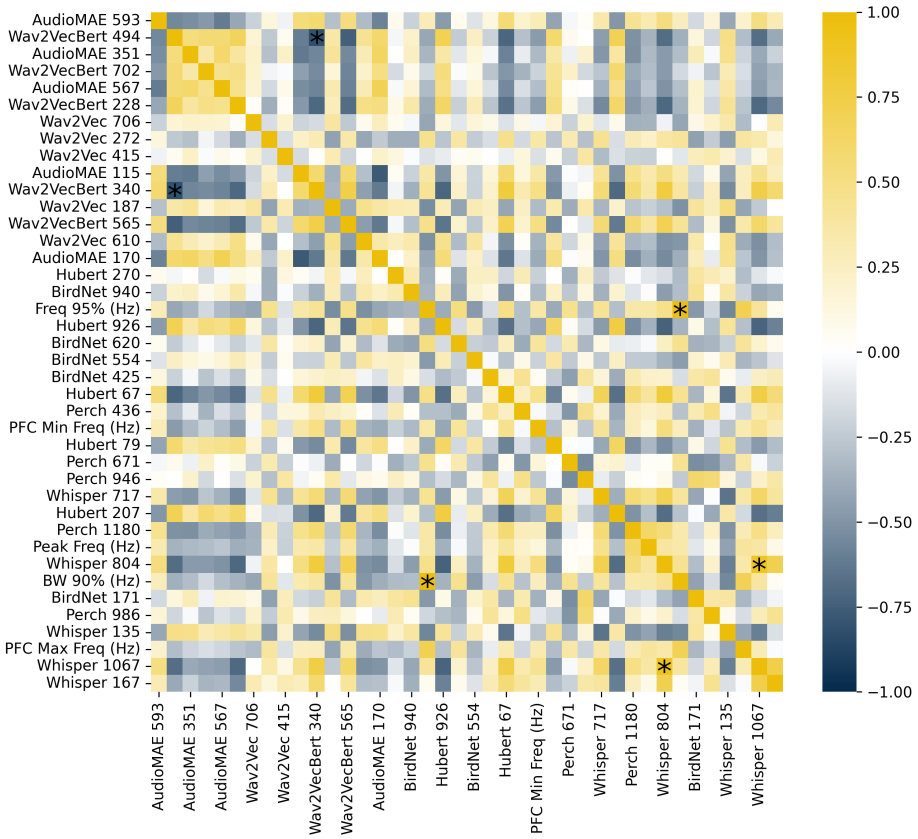


Figure 5: Mutual information of the top features in the Peeps call type dataset, spanning seven acoustic pre-trained embeddings. We display the five highest-performing features per pre-trained embedding, along with the top five interpretable features. Asterisks show correlation coefficients above 0.8.

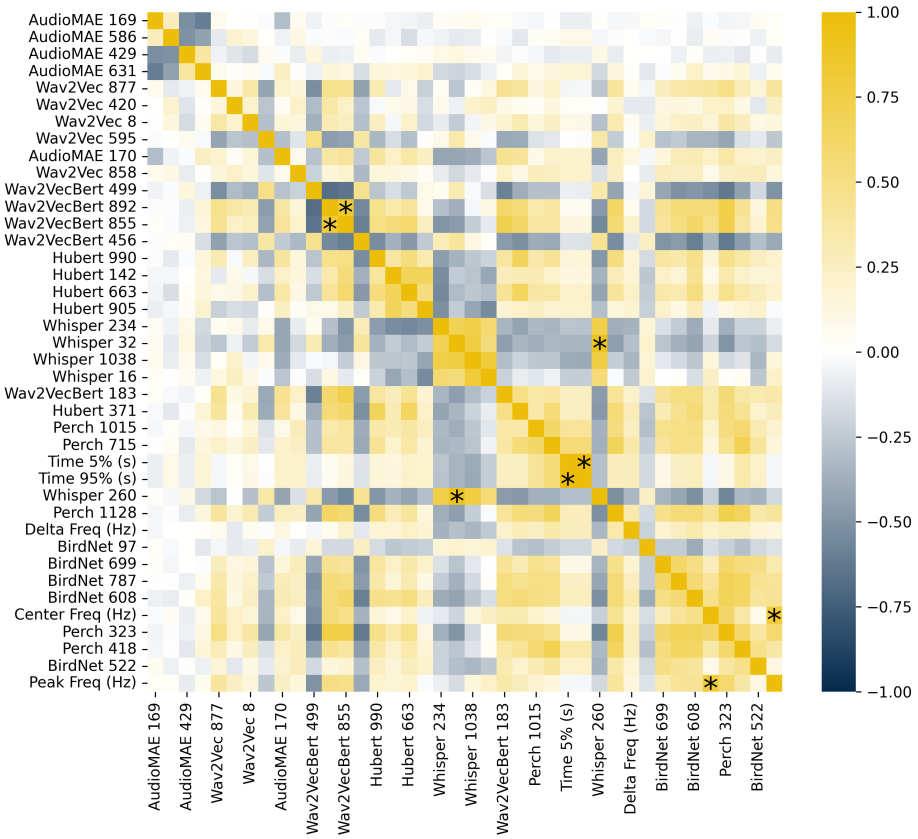


Figure 6: Mutual information of the top features in the Twitters call type dataset, spanning seven acoustic pre-trained embeddings. We display the five highest-performing features per pre-trained embedding, along with the top five interpretable features. Asterisks show correlation coefficients above 0.8.

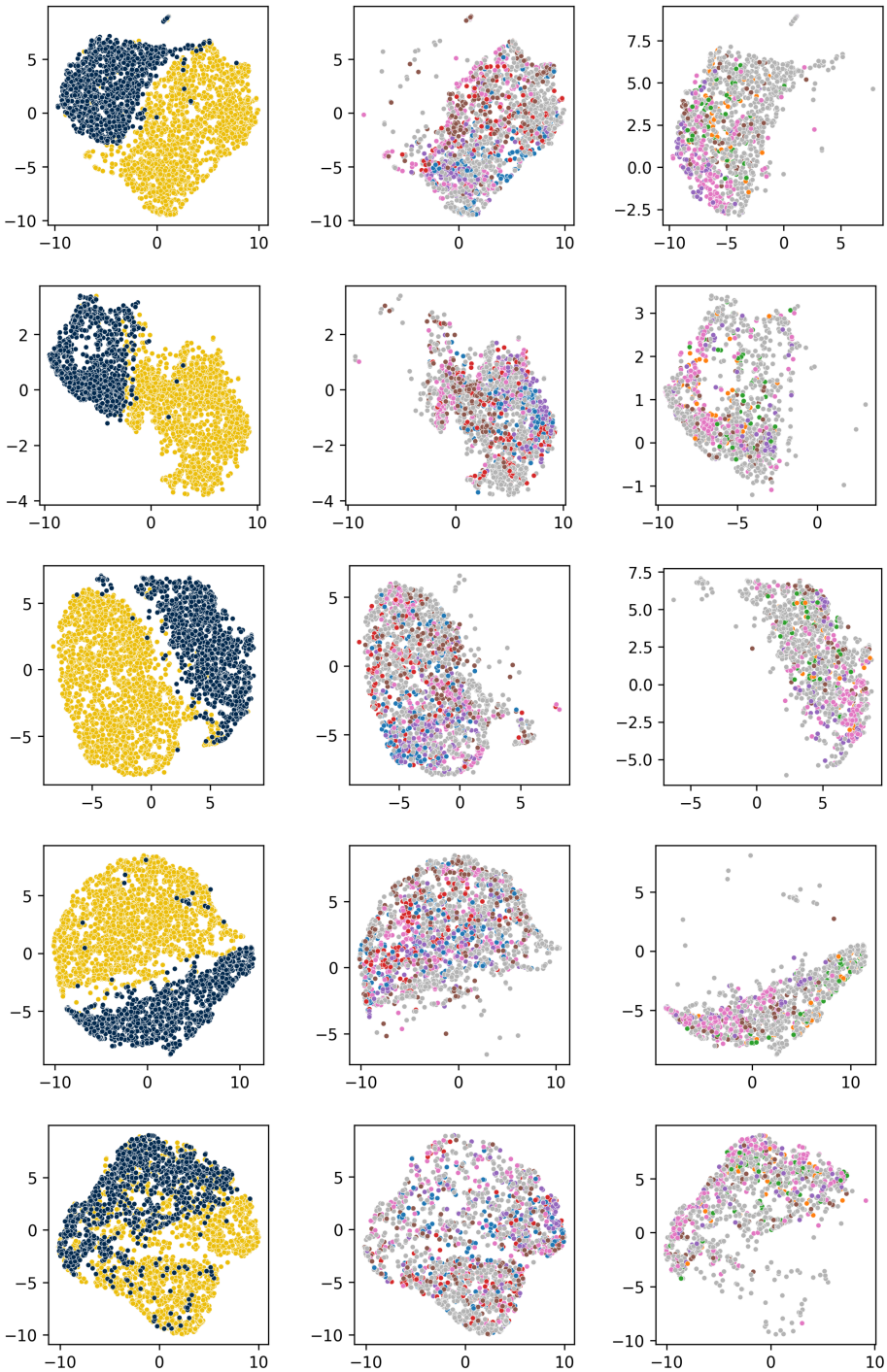


Figure 7: t-SNE visualizations of five pre-trained embeddings, primarily trained on human speech data (with AudioMAE also incorporating internet-sourced audio). The first column presents the t-SNE plot of call types (Peeps in yellow and Twitters in blue), while the second and third columns show the t-SNE projections of Peeps and Twitters, respectively, with points colored by individual identity. From top to bottom, the rows correspond to HuBERT, Wav2Vec, Wav2Vec BERT, Whisper, and AudioMAE.

Model	F1 Score (Mean ± Std)
Simple Network	
Perch	0.66 ± 0.02
Whisper	0.62 ± 0.03
BirdNET	0.60 ± 0.02
HuBERT	0.55 ± 0.02
Wav2Vec2	0.47 ± 0.02
Concatenation (2 Embeddings)	
BirdNET + Perch	0.67 ± 0.02
Perch + Whisper	0.67 ± 0.02
Perch + HuBERT	0.66 ± 0.02
Perch + AudioMAE	0.64 ± 0.02
Perch + Wav2Vec2	0.64 ± 0.02
Summation (2 Embeddings)	
Perch + Whisper	0.68 ± 0.02
Perch + BirdNET	0.67 ± 0.02
Perch + HuBERT	0.66 ± 0.02
BirdNET + Whisper	0.64 ± 0.02
Perch + Wav2Vec2	0.64 ± 0.02
MRMR (2 Embeddings)	
Perch + Whisper	**0.70 ± 0.02**
Perch + BirdNET	0.69 ± 0.02
Perch + HuBERT	0.68 ± 0.02
Perch + Wav2Vec2	0.67 ± 0.02
Perch + Wav2Vec-bert	0.67 ± 0.02

Table 2: Performance of the top 5 models per method on the acoustic identification task using the Peeps dataset (Mean F1 Score ± Standard Deviation).

Model	F1 Score (Mean ± Std)
Simple Network	
Perch	0.61 ± 0.03
BirdNET	0.60 ± 0.04
HuBERT	0.56 ± 0.04
Whisper	0.55 ± 0.03
Wav2Vec-bert	0.43 ± 0.03
Concatenation (2 Embeddings)	
BirdNET + Whisper	0.63 ± 0.03
BirdNET + Perch	0.62 ± 0.03
Perch + Whisper	0.62 ± 0.03
BirdNET + HuBERT	0.62 ± 0.03
Perch + HuBERT	0.61 ± 0.03
Summation (2 Embeddings)	
BirdNET + Whisper	0.63 ± 0.04
Perch + Whisper	0.63 ± 0.03
BirdNET + Perch	0.63 ± 0.03
BirdNET + HuBERT	0.62 ± 0.03
Perch + HuBERT	0.62 ± 0.03
MRMR (2 Embeddings)	
Perch + Whisper	**0.66 ± 0.03**
BirdNET + Whisper	0.65 ± 0.03
Perch + HuBERT	0.64 ± 0.03
BirdNET + Perch	0.64 ± 0.03
Perch + Wav2Vec2	0.64 ± 0.03

Table 3: Performance of the top 5 models per method on the acoustic identification task using the Twitters dataset (Mean F1 Score ± Standard Deviation).

Measurement	Units	Definition
Center Freq	Hz	The frequency that divides the selection into two intervals of equal energy (i.e., the 50th percentile frequency) measured on each spectrogram slice.
Freq 25%	Hz	The 25th percentile frequency (first quartile) measured on each spectrogram slice.
Freq 75%	Hz	The 75th percentile frequency (third quartile) measured on each spectrogram slice.
Freq 5%	Hz	The 5th percentile frequency measured on each spectrogram slice, indicating the lower bound of the energy distribution.
Freq 95%	Hz	The 95th percentile frequency measured on each spectrogram slice, indicating the upper bound of the energy distribution.
BW 50%	Hz	The inter-quartile range bandwidth, computed as the difference between the 75th and 25th percentile frequencies (i.e., the bandwidth containing 50% of the energy).
BW 90%	Hz	The bandwidth encompassing 90% of the signal's energy, calculated as the difference between the 95th and 5th percentile frequencies.
Peak Freq	Hz	The frequency at which the maximum power (or peak power) occurs within the selection, as observed in each spectrogram slice.
Center Time	s	The time that divides the selection into two intervals of equal energy (i.e., the median or 50th percentile time) for the signal's energy distribution.
Time 25%	s	The time by which 25% of the total energy has been accumulated within the selection.
Time 75%	s	The time by which 75% of the total energy has been accumulated within the selection.
Dur 50%	s	The duration over which the central 50% of the signal's energy is distributed, computed as the difference between the 75th and 25th percentile times.
Time 5%	s	The time by which 5% of the total energy has been accumulated within the selection.
Time 95%	s	The time by which 95% of the total energy has been accumulated within the selection.
Dur 90%	s	The duration over which 90% of the signal's energy is distributed, computed as the difference between the 95th and 5th percentile times.
Delta Freq	Hz	The difference between the upper and lower frequency limits of the selection.
Delta Time	s	The difference between the beginning and ending times of the selection.
Time 5% Rel.	–	The relative time (as a proportion of total duration) at which 5% of the signal's energy is accumulated.
Time 25% Rel.	–	The relative time at which 25% of the signal's energy is accumulated.
Center Time Rel.	–	The relative time corresponding to the median (50%) of the signal's energy distribution.
Time 75% Rel.	–	The relative time at which 75% of the signal's energy is accumulated.
Time 95% Rel.	–	The relative time at which 95% of the signal's energy is accumulated.
Peak Time Relative	–	The time at which the peak amplitude occurs, expressed as a proportion of the total selection duration.
PFC Avg Slope	Hz/ms	The average slope of the peak frequency contour over time, computed as the mean of the differences between successive peak frequencies.
PFC Max Freq	Hz	The maximum frequency reached in the peak frequency contour.
PFC Max Slope	Hz/ms	The maximum rate of change (slope) observed in the peak frequency contour.
PFC Min Freq	Hz	The minimum frequency reached in the peak frequency contour.
PFC Min Slope	Hz/ms	The minimum rate of change (slope) observed in the peak frequency contour.
PFC Num Inf Pts	–	The number of inflection points in the peak frequency contour, indicating how frequently the slope changes sign.

Table 4: Summary of acoustic measurements derived from Raven Pro 1.6. Definitions are adapted from the Raven Pro manual.

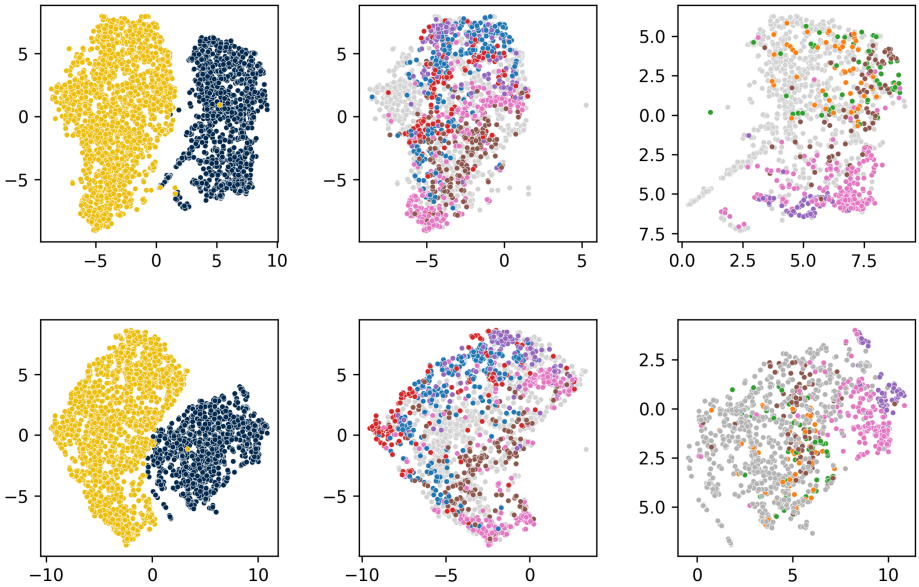


Figure 8: tSNE visualizations of five pre-trained embeddings, primarily trained on bioacoustics bird data (with BirdNET also incorporating other animals). The first column represents the t-SNE plot of call types (Peeps in yellow and Twitters in blue), while the second and third columns depict the t-SNE projections of Peeps and Twitters colored by individual, respectively. From top to bottom, the rows correspond to BirdNET and Perch, respectively.

Model name	**Number of parameters**	**Training data (hours)**	**Reference**
BirdNET	27M	8300	Kahl et al. (2021)
HuBERT-Large	1B	60960	Hsu et al. (2021)
Perch	7.8M	<10k	Ghani et al. (2023b)
Wav2vec2	317M	54000	Baevski et al. (2020)
W2v-BERT 2.0	600M	60960	Hsu et al. (2021)
Whisper-Large-v2	1.55B	680000	Radford et al. (2022)
AudioMAE	304M	5500	Huang et al. (2023)

Table 5: List of considered models for acoustic embeddings, including their size, training data, and references.

SELF-PERCEPT: Introspection Improves Large Language Models' Detection of Multi-Person Mental Manipulation in Conversations

Danush Khanna[1] **Pratinav Seth**[2,3] **Sidhaarth Murali**[4]
Aditya Guru[1] **Siddharth Shukla**[1] **Tanuj Tyagi**[1]
Sandeep Chaurasia[1] **Kripabandhu Ghosh**[5]
[1]Manipal University Jaipur, India [2]Manipal Institute of Technology, India
[3]AryaXAI Alignment Lab, AryaXAI.com, India
[4]National Institute of Technology Karnataka, Surathkal, India [5]IISER Kolkata, India
danush.229310455@muj.manipal.edu

Abstract

Mental manipulation is a subtle yet pervasive form of abuse in interpersonal communication, making its detection critical for safeguarding potential victims. However, due to manipulation's nuanced and context-specific nature, identifying manipulative language in complex, multi-turn, and multi-person conversations remains a significant challenge for large language models (LLMs). To address this gap, we introduce the MultiManip dataset, comprising 220 multi-turn, multi-person dialogues balanced between manipulative and non-manipulative interactions, all drawn from reality shows that mimic real-world scenarios. For manipulative interactions, it includes 11 distinct manipulations depicting real-life scenarios. We conduct extensive evaluations of state-of-the-art LLMs, such as GPT-4o and Llama-3.1-8B, employing various prompting strategies. Despite their capabilities, these models often struggle to detect manipulation effectively. To overcome this limitation, we propose SELF-PERCEPT, a novel, two-stage prompting framework inspired by Self-Perception Theory, demonstrating strong performance in detecting multi-person, multi-turn mental manipulation. Our code and data are publicly available at https://github.com/danushkhanna/self-percept

1 Introduction

Significant progress has been made in natural language processing (NLP) for the early detection of various mental health symptoms, such as stress (Nijhawan et al., 2022), depression (Xu et al., 2019), and suicidal tendencies (Coppersmith et al., 2018), among others (Naseem et al., 2022; Xu et al., 2023). Similarly, efforts in toxic speech detection target both explicit and implicit forms of toxicity (Yavnyi et al., 2023; Miao et al., 2020). Detection of such Toxicity in Multi-turn conversation, especially in multi-person contexts, remains a relatively underexplored problem, similar to the detection of manipulation. However, recent works (Sun, 2024; Senese et al., 2020; Li et al., 2022; Yang et al., 2022) have made progress by proposing various approaches. While NLP techniques have advanced in detecting verbal toxicity, the focus has primarily been on context-free content, leaving implicit manipulation underexplored. In contrast, manipulation is subtle and context-dependent, making it harder to detect. Mental manipulation is a deceptive tactic to control or influence thoughts and emotions for personal gain (Barnhill, 2014). With digital technologies, opportunities for manipulation have grown, impacting both personal interactions and information dissemination, leading to mental health issues (Ienca, 2023). Recent initiatives, such as the development of the MentalManip dataset (Wan et al., 2024), aim to fill this gap. This dataset, which contains annotated fictional dialogues, highlights manipulation techniques and the vulnerabilities they exploit. However, it faces limitations, particularly in evaluating complex, multi-person real-world scenarios. Additionally, it lacks a balanced evaluation set.

Recent work builds on the MentalManip dataset (Ma et al., 2024), as it advances basic detection, the work reduces manipulation detection to a binary classification task ("manipulative" or "non-manipulative"), focusing only on two-person conversations. This neglects more complex multi-party interactions, limiting the ability to capture nuanced manipulative tactics that emerge in group dynamics. These limitations raise a crucial question: ***Can large language models effectively identify various manipulation techniques in complex, multi-turn, multi-participant dialogues that resemble real-world conversations?***

To answer this question, we conducted extensive experiments using large language models, including GPT-4o (Achiam et al., 2023; Murali et al., 2024) and Llama-3.1-8B (Dubey et al., 2024), on a new balanced multi-turn, multi-person mental manipulation detection dataset, applying novel and state-of-the-art prompting such as zero-shot,

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 660–675
July 27 - August 1, 2025

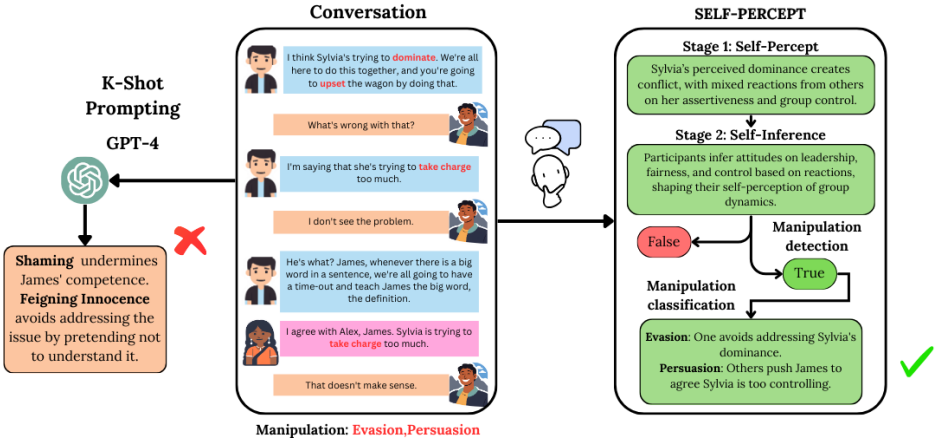


Figure 1: An example of a manipulative conversation from the MultiManip Dataset, including illustrations of the Proposed SELF-PERCEPT Prompting Method and outputs from both SELF-PERCEPT and K-shot GPT-4o.

few-shot, and Chain-of-Thought (Wei et al., 2022) strategies and evaluating them on various metrics (as detailed in Section 3). Our research makes the following contributions:

- We developed **MultiManip**, a novel dataset containing 220 multi-turn, multi-person dialogues from reality shows. This dataset features an balanced representation of manipulative and non-manipulative interactions, reflecting real-world scenarios and encompassing 11 distinct manipulation techniques (Section 2).
- We propose `SELF-PERCEPT`, a novel, two-stage prompting framework that draws from Self-Perception Theory to enhance manipulation detection in dynamic conversations, improving model accuracy and interpretability in real-world settings (See Section 3).
- We performed extensive evaluations of LLMs, using various prompting methods to assess their effectiveness in multi-person manipulation detection (detailed in Section 3).

2 MultiManip

We introduce MultiManip, a dataset designed to assess how well models and systems can identify manipulation in real-world conversations involving both manipulative and non-manipulative conversations in balanced distribution with multiple persons, and multi-label manipulations to depict real-world scenarios. The taxonomy of manipulation techniques, as depicted in Table 3, is adapted from psychological research (Barnhill, 2014) for conversational contexts. The taxonomy aligns with MentalManip (Wan et al., 2024) but extends to multi-turn, multi-person interactions in MultiManip.

There is a clear shortage of publicly available datasets focused specifically on mental manipulation in conversations. While MentalManip (Wan et al., 2024) is one of the first datasets for mental manipulation, its creation is based on the Cornell movie dataset (Danescu-Niculescu-Mizil and Lee, 2011). Movies often follow certain tropes and may not reflect everyday interactions accurately. While we acquire our data from reality shows, which have instances much more similar to real-life instances of conversations that might have a multi-person, multi-turn manipulation setup instead of a direct conversation between two people.

MultiManip (ours) advances MentalManip in two ways: (1) Multi-person interactions (vs. dyadic) (2) Real-world context (reality shows vs. fictional scripts). These features better capture real-world dynamics, where manipulation emerges through group negotiation rather than isolated exchanges. Please refer Appendix E.

2.1 Data Source and Pre-Processing

To address the challenges of multi-turn conversations and realistic dialogue that may involve real-life scenarios, we extract transcripts from the Fandom[1] website featuring the Survivor TV Series [2], which is a publicly available source. This show captures genuine interactions between contestants, offering authentic examples of manipulative behavior. The competitive nature of the series often leads participants to use strategic and manipulative tactics,

[1] https://www.fandom.com/
[2] https://www.imdb.com/title/tt0239195/

making it a valuable source of relevant data. Additionally, the diversity of contestants from various backgrounds allows us to observe a broad range of manipulative techniques and responses. We gather transcripts from multiple seasons of the TV series. Since manipulative language is relatively rare in everyday conversation, we first need to filter the raw data to identify dialogues that might contain manipulative elements. To achieve this, we use a large language model (LLM), specifically Llama-3.1-70B, in a zero-shot approach. This model is employed to carry out a multi-step analysis, focusing on extracting specific conversational patterns to detect manipulation techniques within the data. We employed a multi-model approach to mitigate LLM bias during extraction: Llama-3.1-70B for initial filtering and GPT-4o/Llama-3.1-8B for evaluation. Manual verification ensured diverse representation across manipulation types. The dataset's balanced structure minimizes class imbalance, as detailed in Appendix A.3. Future work will expand on the sample size and cultural contexts. We detail on the data preprocessing and prompt templates in Appendix A.2.

2.2 Human Annotation and Label Curation

We recruited five college students, all of whom are native or fluent English speakers, to serve as annotators. (More details in the Section 6). In each task, annotators are shown a dialogue and asked to answer two questions for each instance: ($\mathcal{Q}1$) [Binary Choice: 'Yes" or "No"]: Does it include any elements of mental manipulation? ($\mathcal{Q}2$) [Multiple Choice: Options in Table 3.]: What manipulation techniques are used? $\mathcal{Q}2$ is answered only if the response to $\mathcal{Q}1$ is "Yes."

Final labels were derived by aggregating annotations from five annotators per dialogue, using majority rule in cases of disagreement. Fleiss' Kappa, an extension of Cohen's Kappa for multiple raters, was used to assess annotator consistency, yielding a score of 0.429, indicating moderate agreement (0.41–0.60). Overall, the annotators found the task to be of a challenging nature. Non-manipulative texts were felt to be comparatively easier to annotate with more consensus, while manipulative texts involving anger, shaming, and persuasion were more straightforward. However, annotators found accusation, denial, and feigning innocence challenging. Detailed data on Median Agreement and Median Agreement Score are available in Appendix Table 6. Further feedback on these difficulties is detailed in Appendix A.4 and A.5.

3 Methodology

3.1 Motivation

Understanding the underlying attitudes and beliefs of individuals in a conversation is crucial for a variety of tasks, such as sentiment analysis, stance detection, and dialogue understanding. Conventional prompting techniques like CoT often struggle to capture the subtle cues that signal these internal states. Inspired by Self-Perception Theory (SPT) which suggests that individuals infer their own attitudes by observing their behaviors, we propose our novel prompting framework, SELF-PERCEPT (Fazio, 2014; Woosnam et al., 2018; Ross and Shulman, 1973; Calder and Staw, 1975; Haemmerlie and Montgomery, 1982). This framework leverages principles from SPT to help Large Language Models (LLMs) systematically interpret character behaviors and self-inferences, enhancing their ability to analyze and infer attitudes within conversational contexts, Please refer Appendix D. Unlike Chain-of-Thought prompting, which focuses on stepwise reasoning, SELF-PERCEPT's Stage 1 explicitly extracts behavioral cues (verbal/non-verbal) to infer latent attitudes. For example, detecting a sigh (non-verbal) alongside agreement (verbal) reveals passive-aggressive intent. This aligns with Self-Perception Theory (Fazio, 2014), where behaviors inform internal states. Stage 2 then synthesizes these observations, enabling nuanced detection in multi-person settings. By guiding LLMs in this way, SELF-PERCEPT aims to improve reasoning in complex NLP tasks.

3.2 SELF-PERCEPT

SELF-PERCEPT is a two-stage prompting framework that mirrors the process of behavioral observation and self-inference, enabling LLMs to dissect conversations effectively and infer personal attitudes based on observed behaviors (Image 1). Please refer to prompts templates detailed in Appendix C and output stages in Appendix B.1.

3.2.1 Stage 1: Self-Percept

We first begin with holistically observing and analyzing the behaviors of multiple participants within conversations, including both verbal and non-verbal cues. (C.f Appendix C) This stage emphasizes identifying discrepancies between words and actions, thereby understanding the behavior of

Model	Prompting Method	Acc.	P	R	F_1
GPT-4o	Zero-Shot	0.27	0.20	0.31	0.16
	Few-Shot	0.39	0.19	0.21	0.22
	CoT	0.34	0.21	**0.32**	0.34
	SELF-PERCEPT (Proposed)	**0.42**	**0.31**	0.20	**0.37**
Llama 3.1 8B	Zero-Shot	0.11	0.09	**0.37**	0.29
	Few-Shot	0.22	0.17	0.36	0.13
	CoT	0.28	**0.23**	0.26	0.10
	SELF-PERCEPT (Proposed)	**0.30**	0.17	0.26	**0.34**

Table 1: Results of multi-label manipulation detection task on MultiManip Dataset. P, R, $Acc.$ and F_1 stand for Precision, Recall, Macro F1 Scores and Accuracy (best results in **bold**)

Model	Prompting Method	Acc.	P	R	F_1
GPT-4o	Zero-Shot	0.11	0.3	**0.62**	0.38
	Few-Shot	0.22	**0.39**	0.53	0.39
	CoT	0.35	0.37	0.56	0.43
	SELF-PERCEPT (Proposed)	**0.45**	0.34	0.55	**0.47**
Llama 3.1 8B	Zero-Shot	0.02	0.11	**0.56**	0.17
	Few-Shot	0.04	0.07	0.35	0.11
	CoT	0.19	0.14	0.38	0.18
	SELF-PERCEPT (Proposed)	**0.23**	**0.21**	0.32	**0.19**

Table 2: Results of multi-label manipulation detection task on Test Set of MentalManip Dataset. P, R, $Acc.$ and F_1 stand for Precision, Recall, Macro F1 Scores and Accuracy (best results in **bold**).

all participants in the conversation. The process highlights potential contradictions and manipulative techniques. The output is a list of observed behaviors and statements, which forms the foundation for further inference. (C.f Appendix B.1)

3.2.2 Stage 2: Self-Inference

Using the observations from Stage 1, the model infers characters' internal attitudes or beliefs, paying special attention to whether manipulation is detected and specifying the type of manipulation based on predefined categories. This concise inference is intended to capture the essence of interpersonal dynamics. (C.f Appendix B.1)

3.3 Result and Analysis

The results from the `MultiManip Dataset` and the `MentalManip Dataset` highlight the effectiveness of `SELF-PERCEPT` in multi-label manipulation detection. Across both datasets, `SELF-PERCEPT` consistently outperforms baseline methods (Zero-Shot, Few-Shot, and CoT) in both `GPT-4o` (C.f Table 7) and `Llama 3.1 8B` (C.f Table 8) models, providing a balanced combination of Precision and Recall.

On the `MultiManip Dataset` (C.f. Table 1), `SELF-PERCEPT` achieves the highest Accuracy (0.42) and F1 score (0.37) in `GPT-4o`, outperforming all other methods. While `SELF-PERCEPT` has slightly lower Recall (0.20) compared to CoT (0.32), it compensates with a higher Precision (0.31), resulting in better overall detection. In `Llama 3.1 8B`, `SELF-PERCEPT` again leads in F1 score (0.34) and Accuracy (0.30), offering a better balance of Precision (0.17) and Recall (0.26) than other methods.

For the `MentalManip Dataset` (C.f. Table 2), `SELF-PERCEPT` delivers the best performance in `GPT-4o` with an Accuracy of 0.45 and F1 score of 0.47. It outperforms Zero-Shot, which has high Recall (0.62) but low Precision (0.30), resulting in a lower F1 score (0.38). In `Llama 3.1 8B`, `SELF-PERCEPT` provides a reasonable balance with Precision (0.21) and Recall (0.32), outperforming Zero-Shot (0.11 Precision) and Few-Shot (0.07 Precision). The modest performance gains (e.g., SELF-PERCEPT's +5% F1 over CoT) stem from its structured behavioral analysis (C.f Appendix B), which reduces false positives by contextualizing intent. For instance, GPT-4o's higher precision (0.31 vs. CoT's 0.21) reflects improved discernment of benign vs. manipulative rationalizations. The dataset's size limits statistical power, but trends suggest scalability with larger data.

The SHAP value plots shown in (Figure 3) illustrate the contribution of individual words towards the classification decision for manipulation for a given dialogue (C.f Appendix B.2). In these plots, red bars indicate a positive SHAP value, contributing towards a "No" (non-manipulative) classification, while blue bars indicate negative SHAP values, contributing towards a "Yes" (manipulative) classification. The SPT Stage 1 model correctly identifies manipulation, as evidenced by the strong negative SHAP values for words like "anxious," "situation," and "teamwork", which capture psychological pressure, power imbalance, and persuasive intent (C.f Appendix B.3). These attributes increase the negative influence on the "No" classification, effectively outweighing the overall SHAP score towards a final "Yes" decision (manipulation detected). Conversely, CoT misclassifies the dialogue as non-manipulative due to an over-reliance on neutral or goal-driven words such as "game," "desire," and "focused", which lack contextual awareness of social influence. As a result, CoT assigns higher positive SHAP values to these words, failing to capture the subtleties of persua-

sion, thereby skewing its decision towards "No". This contrast highlights SPT's superior ability to detect manipulation by appropriately weighting behavioral attributes indicative of coercion and dominance, while CoT overlooks these critical cues.

4 Conclusion

This study introduces the MultiManip dataset to enhance mental manipulation detection in complex, real-world conversations with multi-turn, multi-participant dialogues. Our evaluation of leading LLMs across various benchmarks revealed challenges in detecting manipulation in these dynamic settings.

To address this, we propose the SELF-PERCEPT framework, a two-stage prompting method that improves detection through self-observation and self-inference. It shows notable improvements in the F1 score, effectively balancing Precision and Recall for both GPT-4o and Llama-3.1-8B models. The framework's multi-step approach, especially its self-perception stage, reduces false positives and negatives, making it a robust solution for manipulation detection.

Future work should refine these methods and apply them to diverse social media platforms, dialects, and languages. (C.f Appendix F, F.1) Additionally, expanding the size of the dataset will be prioritized to further improve detection accuracy.

5 Limitations

Despite advancements with the MultiManip dataset and the SELF-PERCEPT framework, some limitations remain. The MultiManip dataset, based on transcripts from the Survivor TV Series on Fandom, provides real examples of manipulative behavior in competitive, multi-turn dialogues. However, it may not fully capture the range of real-world conversational nuances where manipulation can be even less predictable. While it includes genuine interactions with various contestants, its focus on a specific context might limit its relevance to broader, more unstructured conversations. Additionally, the dataset is relatively small, with only 220 samples representing 11 manipulation techniques. This size is insufficient for large-scale model training and is mainly useful for evaluation. The dataset might also lack diverse manipulative scenarios and cultural contexts that could impact the model's generalizability. Moreover, current leading language models like GPT-4o and Llama-3.1-8B have difficulty detecting manipulation in complex, multi-turn conversations, suggesting these models still lack the necessary contextual understanding and inference skills. Implementing and scaling the SELF-PERCEPT framework in real-world scenarios could be challenging (C.f Appendix F.2, F.3). We provide recommendations for responsible use detailed in Appendix F.4. The two-stage prompting approach may require substantial computational resources and fine-tuning to adapt to various conversational contexts and detection needs. Addressing these issues is essential for improving manipulation detection and making these methods more applicable in diverse real-world situations.

6 Annotation Acknowledgement

We recruited 5 undergraduate students who are native or fluent English speakers as annotators. The group represented diversity in terms of gender (two females, three males), ethnicity, educational, and cultural backgrounds. Preference was given to applicants with experience in psychology or linguistics during the selection process.

To ensure annotators understood the techniques before starting, we held training sessions, asked them to carefully review the instructions, and closely monitored their work. The training included examples and illustrations to explain what counts as manipulation. We also gave concrete examples of different manipulation techniques and showed how multiple techniques could appear in one conversation. This preparation helped them label the dialogues accurately and with confidence. Plus, we provided a reference manual with definitions of each technique to guide them through the process.

We acknowledge that annotating manipulation and its various techniques can be challenging (For more details about feedback from Annotators, refer to Appendix A). All annotators were compensated for their efforts, following institutional guidelines, with a recommendation to limit themselves to no more than 20 samples per day to prevent fatigue. Dataset transcripts are anonymized (e.g., "Person A") and sourced from publicly available materials, avoiding privacy concerns.

References

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Alt-

man, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haim ing Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Benjamin Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Sim'on Posada Fishman, Juston Forte, Is abella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Raphael Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Lukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Ryan Kiros, Matthew Knight, Daniel Kokotajlo, Lukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Ma teusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel P. Mossing, Tong Mu, Mira Murati, Oleg Murk, David M'ely, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Ouyang Long, Cullen O'Keefe, Jakub W. Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alexandre Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Pondé de Oliveira Pinto, Michael Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack W. Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario D. Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin D. Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas A. Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cer'on Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll L. Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qim ing Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report. *None*.

Anne Barnhill. 2014. What is manipulation? In *Manipulation: Theory and Practice*. Oxford University Press.

Bobby J Calder and Barry M Staw. 1975. Self-perception of intrinsic and extrinsic motivation. *Journal of personality and social psychology*, 31(4):599.

Glen Coppersmith, Ryan Leary, Patrick Crutchley, and Alex Fine. 2018. Natural language processing of social media as screening for suicide risk. *Biomedical Informatics Insights*, 10:1178222618792860.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Cantón Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab A. AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriele Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guanglong Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet,

Jaewon Lee, Jan Laurens Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Ju-Qing Jia, Kalyan Vasuden Alwala, K. Upasani, Kate Plawiak, Keqian Li, Ken-591 neth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline C. Muzzi, Mahesh Babu Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melissa Hall Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri S. Chatterji, Olivier Duchenne, Onur cCelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Chandra Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yiqian Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zhengxu Yan, Zhengxing Chen, Zoe Papakipos, Aaditya K. Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adi Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Ben Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Shang-Wen Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzm'an, Frank J. Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory G. Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kaixing(Kai) Wu, U KamHou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollár, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sung-Bae Cho, Sunny Virk, Suraj Subramanian,

Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Andrei Poenaru, Vlad T. Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xia Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

Russell H Fazio. 2014. Self-perception theory: A current perspective. *Social Influence*, pages 129–150.

Frances M Haemmerlie and Robert L Montgomery. 1982. Self-perception theory and unobtrusively biased interactions: A treatment for heterosocial anxiety. *Journal of Counseling Psychology*, 29(4):362.

Marcello Ienca. 2023. On artificial intelligence and manipulation. *Topoi*, 42(3):833–842.

Jiatong Li, Bin He, and Fei Mi. 2022. Exploring effective information utilization in multi-turn topic-driven conversations. *ArXiv*, abs/2209.00250.

Jiayuan Ma, Hongbin Na, Zimu Wang, Yining Hua, Yue Liu, Wei Wang, and Ling Chen. 2024. Detecting conversational mental manipulation with intent-aware prompting. *Preprint*, arXiv:2412.08414.

Lin Miao, Mark Last, and Marina Litvak. 2020. Detecting troll tweets in a bilingual corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6247–6254, Marseille, France. European Language Resources Association.

Sidhaarth Murali, S Sowmya, and R Supreetha. 2024. Remag-kr: Retrieval and medically assisted generation with knowledge reduction for medical question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 62–67.

Usman Naseem, Adam G. Dunn, Jinman Kim, and Matloob Khushi. 2022. Early Identification of Depression Severity Levels on Reddit Using Ordinal Classification. In *Proceedings of the ACM Web Conference 2022*, WWW '22, pages 2563–2572, New York, NY, USA. Association for Computing Machinery.

Tanya Nijhawan, Girija Attigeri, and T. Ananthakrishna. 2022. Stress detection using natural language processing and machine learning over social interactions. *Journal of Big Data*, 9(1):33.

Michael Ross and Ronald F Shulman. 1973. Increasing the salience of initial attitudes: Dissonance versus self-perception theory. *Journal of Personality and Social Psychology*, 28(1):138.

Matteo Antonio Senese, Giuseppe Rizzo, Mauro Dragoni, and Maurizio Morisio. 2020. MTSI-BERT: A session-aware knowledge-based conversational agent. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 717–725, Marseille, France. European Language Resources Association.

Yiping Sun. 2024. A global-local attention mechanism for relation classification. *ArXiv*, abs/2407.01424.

Yuxin Wan, Ivory Yang, Saeed Hassanpour, and Soroush Vosoughi. 2024. Mentalmanip: A dataset for fine-grained analysis of mental manipulation in conversations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3747–3764.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Kyle M Woosnam, Jason Draper, Jingxian Kelly Jiang, Kayode D Aleshinloye, and Emrullah Erul. 2018. Applying self-perception theory to explain residents' attitudes about tourism development through travel histories. *Tourism Management*, 64:357–368.

Xuhai Xu, Prerna Chikersal, Afsaneh Doryab, Daniella K. Villalba, Janine M. Dutcher, Michael J. Tumminia, Tim Althoff, Sheldon Cohen, Kasey G. Creswell, J. David Creswell, Jennifer Mankoff, and Anind K. Dey. 2019. Leveraging Routine Behavior and Contextually-Filtered Features for Depression Detection among College Students. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, volume 3, pages 116:1–116:33.

Xuhai Xu, Bingsheng Yao, Yuanzhe Dong, Saadia Gabriel, Hong Yu, James Hendler, Marzyeh Ghassemi, Anind K. Dey, and Dakuo Wang. 2023. Mental-llm: Leveraging large language models for mental health prediction via online text data. *arXiv preprint arXiv:2307.14385*.

Yang Yang, Juan Cao, Yujun Wen, and Pengzhou Zhang. 2022. Multiturn dialogue generation by modeling sentence-level and discourse-level contexts. *Scientific Reports*, 12.

Serhii Yavnyi, Oleksii Sliusarenko, Jade Razzaghi, Olena Nahorna, Yichen Mo, Knar Hovakimyan, and Artem Chernodub. 2023. DeTexD: A Benchmark Dataset for Delicate Text Detection. In *The 7th Workshop on Online Abuse and Harms (WOAH)*, pages 14–28, Toronto, Canada. Association for Computational Linguistics.

A Dataset Annotation

A.1 Inter-Annotator Reliability

Fleiss' Kappa is a statistical measure used to evaluate the reliability of agreement between multiple raters who assign categorical labels to a set of items. It generalizes Cohen's Kappa, which is applicable for two raters, to accommodate any number of raters.

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e}$$

Where:

$\bar{P}$ is the mean of the observed agreement over all items, defined as:

$$\bar{P} = \frac{1}{N} \sum_{i=1}^{N} P_i$$

P_i is the observed agreement for the i-th item, calculated as:

$$P_i = \frac{1}{n(n-1)} \sum_{j=1}^{k} n_{ij}(n_{ij} - 1)$$

$\bar{P}_e$ is the expected agreement by chance, calculated as:

$$\bar{P}_e = \sum_{j=1}^{k} p_j^2$$

Where: N is the number of items, n is the number of raters per item, k is the number of categories, n_{ij} is the number of raters who assigned category j to item i, p_j is the proportion of all assignments to category j, defined as:

$$p_j = \frac{1}{Nn} \sum_{i=1}^{N} n_{ij}$$

To measure inter-annotator agreement, we employed Fleiss' Kappa, which is an extension of Cohen's Kappa for multiple annotators. The dataset used consists of 5 annotators, where each annotator could assign one or more labels from a predefined set of 12 possible labels to each item. The 12 labels include 11 manipulation methods and 1 for non-manipulative. The resulting Fleiss' Kappa value was **0.429**, indicating moderate agreement (0.41 - 0.60) between the annotators.

A.2 Dataset Curation and Pre-Processing

To ensure effective processing, the transcripts are divided into smaller chunks of 10,000 characters, with a 2,000-character overlap between sections to maintain context across adjacent chunks. Non-manipulative data is processed similarly. After completing this step, we extracted the usable data in CSV format and labeled it for further steps like preprocessing, data cleaning, and human annotation. To ensure the accurate segmentation of text chunks, we manually verify and remove any irrelevant parts before proceeding with data cleaning and human annotation.

A.2.1 Prompt for Manipulative Sample

Manipulation Prompting format:

```
'''
First, print the watermark before every
extraction: "*".
You must identify and extract conversations
that involve the following manipulation
technique:
<Manipulation Method>:<Manipulation
 Definition>.

- The extracted conversations must
include 3 participants.
- The extracted conversations must be
multi-turn, with each participant speaking
more than once.
- Structure the output as follows:

Person A: [Dialogue]
Person B: [Dialogue]
Person C: [Dialogue]
Person B: [Dialogue]
Person A: [Dialogue]
Person C: [Dialogue]

Ensure that the conversation includes at
least one instance of <Manipulation Method>
from one person towards another.
'''
```

A.2.2 Prompt for Non- Manipulative Sample

Non-Manipulation Prompting format:

```
"""
First, print the watermark before every
extraction: "***".
You must identify and extract conversations
that do not involve any manipulation
techniques.

- The extracted conversations must include
3 participants.
- The extracted conversations must be
multi-turn, with each participant speaking
more than once.
- Structure the output as follows:

Person A: [Dialogue]
Person B: [Dialogue]
Person C: [Dialogue]
Person B: [Dialogue]
```

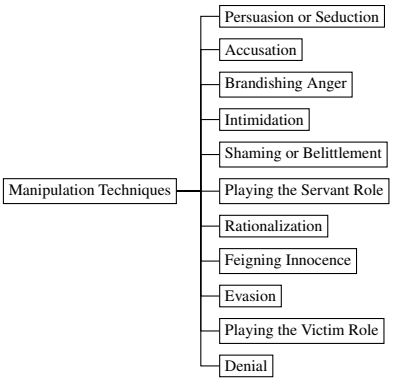


Figure 2: Taxonomy of various Mental Manipulation Techniques. Description of Techniques in Table 3.

```
Person A: [Dialogue]
Person C: [Dialogue]

Ensure that the conversation does not
include any instances of manipulation.
"""
```

A.3 Dataset Distribution

The MultiManip dataset has representation of both manipulative and non-manipulative dialogues. This distribution mirrors real-world conversations, where manipulation can be both subtle and explicit, and its detection depends on context. By balancing manipulative and non-manipulative texts, the dataset enables more accurate training and evaluation of models, helping them distinguish between normal interactions and manipulative behavior across diverse scenarios.

The dataset includes 11 distinct manipulation techniques, ranging from overt tactics like intimidation to subtler methods like denial or playing the victim. This variety ensures that models can detect a wide range of manipulative strategies, rather than focusing on a narrow set of behaviors.

This balanced approach supports the creation of robust models capable of identifying manipulation in multiple forms and contexts. It also prevents bias that could arise from overrepresentation of certain classes, which could lead to overfitting. By maintaining this balance, we promote the development of models that generalize well across different conversational settings.

The balanced structure also reflects the nuanced nature of manipulation, as real-world interactions often blend genuine conversation with manipulative tactics. By capturing this complexity, the dataset mirrors the varied and multi-turn nature of manipulation in everyday life.

A.4 Inferences drawn from Annotation

The final labels were determined by combining annotations from five reviewers per dialogue, with the majority rule applied to resolve disagreements. In cases of differing opinions, Fleiss' Kappa, an extension of Cohen's Kappa for multiple raters, was used to measure annotator consistency, resulting in a score of 0.429, which indicates moderate agreement (0.41–0.60).

Here, the majority rule means selecting the label(s) that appeared most frequently for each instance. If two or more labels had the same highest frequency, multiple labels were chosen, treating the task as a multi-label problem. This approach ensures that all relevant labels are accounted for, acknowledging the complexity of the annotation process when no single label is clearly dominant.

Our analysis based on inference from Table 6 shows that for non-manipulative texts, the majority of cases (at least 3 out of 5 annotators) reached an agreement. A similar trend was observed for categories like anger, persuasion/seduction, and intimidation. However, classes such as accusation, feigned innocence, and denial tended to have lower average agreement among the annotators.

A.5 Feedback from Annotators

The feedback from annotators emphasized the crucial role of balanced distribution between non-manipulative and manipulative samples in the dataset. Non-manipulative dialogues provide a clear baseline, helping annotators distinguish manipulation from normal conversation. Without sufficient non-manipulative examples, annotators risk overestimating manipulation, leading to false positives.

Additionally, an balanced distribution ensures the dataset reflects the real-world prevalence of manipulation, where non-manipulative interactions are far more common. This balance improves consistency in labeling and prevents bias in model training. It also supports model generalization, ensuring that models trained on this data can accurately distinguish manipulation in diverse conversational contexts.

According to the annotators, manipulation techniques involving evasion and feigning innocence were generally easier to identify, while those related to intimidation posed a particular challenge. The dialogues presented in Table 5 illustrate instances where analysis was especially difficult due

Manipulation Method	Definition
Denial	The manipulator denies any wrongdoing or pretends to be confused about others' concerns.
Playing the Victim Role	The manipulator portrays themselves as a victim to gain sympathy, attention, or to divert focus from their own misconduct.
Evasion	The manipulator avoids addressing an issue by giving irrelevant or vague responses.
Feigning Innocence	The manipulator implies that any harm caused was accidental, downplaying their responsibility.
Rationalization	The manipulator justifies their inappropriate behavior with excuses or seemingly logical explanations.
Playing the Servant Role	The manipulator disguises their self-serving actions as contributions to a noble cause.
Shaming or Belittlement	The manipulator uses sarcasm, criticism, or put-downs to make others feel inferior, unworthy, or embarrassed.
Intimidation	The manipulator places others on the defensive by using veiled threats.
Brandishing Anger	The manipulator uses anger to shock the victim into submission by displaying intense emotional responses.
Accusation	The manipulator accuses the victim of being at fault, selfish, uncaring, or living an excessively easy life.
Persuasion or Seduction	The manipulator uses charm, emotional appeal, or logical reasoning to lower the victim's defenses.

Table 3: Definitions of the 11 manipulation techniques.

Person	Conversation (Dialogue)
Person 1	I do believe the reasons are starting to change.
Person 2	What do you mean by that?
Person 1	I really think before, some of the decisions were pretty quick.
Person 2	And now?
Person 1	But I think now people are starting to get to know one another.
Person 1	I think it's about the chemistry.
Person 2	So it's not just about physical strength anymore?
Person 1	No, it's not.
Person 3	My credo in life is love many, trust few, do wrong to none.
Person 2	And how does that apply to the game?
Person 3	So the question is what is few?
Person 2	And how do you decide who to trust?
Person 3	Whether I trust them depends on whether my strategy dovetails with theirs or not."

Table 4: Here is an example of an conversation which was easy to annotate for annotators.

Person	Conversation (Dialogue)
Person 1	I'm telling you.
Person 2	Dude, do I look like I want to keep losing?
Person 3	I'm just asking you, bro.
Person 2	Do you want to ask me that question again?
Person 1	You're bickering with me for no reason.
Person 3	I'm not bickering with you. Dude, I'm not bickering.

Table 5: Here is an example of an conversation which was Hard to annotate for annotators.

Manipulation Techniques	Count	Median Agreement	Mean Agreement Score
Non-manipulation (N_M)	105	3	3.02
Denial (DEN)	10	2	1.90
Evasion (EVA)	9	2	2.00
Feigning Innocence (FEI)	9	2	1.78
Rationalization (RAT)	12	2	2.00
Playing the Victim Role (VIC)	10	2	2.30
Playing the Servant Role (SER)	10	3	2.50
Shaming or Belittlement (S_B)	9	2	2.22
Intimidation (INT)	10	2	2.30
Brandishing Anger (B_A)	9	3	2.77
Accusation (ACC)	10	2	1.80
Persuasion or Seduction (P_S)	10	3	2.60

Table 6: Median Agreement and Mean Agreement Scores for Manipulation Techniques

to the convergence of multiple manipulative strategies. In such cases, annotators noted that social-emotional reasoning (SER), accusations, and displays of anger were important signals. However, they stressed that the emotional context of these conversations should also be taken into account.

Additionally, the annotators observed that distinguishing between intimidation and rationalization was sometimes confusing. Intimidation can often appear as an attempt to apply logic to justify actions, making it difficult to detect unless an explicit threat is present.

By contrast, the conversations shown in Table 4 seemed more authentic rather than examples of rationalization. Even though they technically fit the definition of rationalization, the conversational tone felt more sincere, complicating their categorization as such.

Lastly, the annotators remarked that detecting anger as a form of manipulation is particularly challenging in text. Since this type of manipulation is often conveyed through vocal cues, it can be difficult to identify solely from text-based interactions.

B SELF-PERCEPT

B.1 SELF-PERCEPT Outputs

Input Conversation

```
Person A : You can't give up, bro.
Person B : Who says I gave up?
Person C : If you get knocked back, you stand back up and you
take another knock in the mouth.
Person A : And you know- and I don't want to start another
argument between you two guys- but at times- and I said this to
you, Anthony- I felt that in his own twisted way, Rocky was
actually trying to nudge you to fight back a little more.
Person C : You act like a little girl sometimes, okay? You're
effeminate.
Person B : This is why I don't say anything. This is why I
don't say anything.
```

SPT-Stage 1 Output (GPT-4o)

```
Context: The conversation involves three people, Person A,
Person B, and Person C. They are discussing Person B's attitude
towards facing challenges and possibly their reaction to a
situation where they felt discouraged. Person A seems to be
trying to mediate or provide encouragement, while Person C
appears to be more confrontational. Person B seems defensive
and reluctant to engage.
Question: What behaviors and statements indicate the attitudes
or beliefs of each character? List them clearly.
Person A: - Statement: "You can't give up, bro."
Behavior/Attitude: Encouraging, supportive, believes in
persistence. - Statement: "And you know- and I don't want to
start another argument between you two guys- but at times- and
I said this to you, Anthony- I felt that in his own twisted
way, Rocky was actually trying to nudge you to fight back a
little more." Behavior/Attitude: Mediating, cautious about
causing conflict, believes in the importance of fighting back.
Person B: - Statement: "Who says I gave up?" Behavior/Attitude:
Defensive, denies giving up, possibly sensitive to criticism. -
Statement: "This is why I don't say anything. This is why I
don't say anything." Behavior/Attitude: Frustrated, feels
misunderstood or unfairly judged, indicates a tendency to
withdraw from confrontation.
Person C: - Statement: "If you get knocked back, you stand back
up and you take another knock in the mouth." Behavior/Attitude:
Confrontational, believes in resilience and toughness,
possibly unsympathetic. - Statement: "You act like a little
girl sometimes, okay? You're effeminate." Behavior/Attitude:
Critical, uses gendered insult to demean, believes in
traditional notions of masculinity.
Inconsistencies: - Person A's statement about not wanting to
start another argument but then bringing up a potentially
contentious point suggests a conflict between their desire to
mediate and their need to address the issue. - Person B's
statement "Who says I gave up?" contrasts with their later
statement "This is why I don't say anything," indicating an
internal conflict between defending themselves and feeling
discouraged from speaking up.
Persuasive Techniques: - Person A uses encouragement and
mediation to persuade Person B to adopt a more resilient
attitude. - Person C uses confrontation and insults to provoke
a reaction from Person B, attempting to challenge them into
changing their behavior.
```

SPT-Stage 2 Output (GPT-4o)

```
Manipulation Detected - Yes
Manipulation Type - S_B
```

B.2 Dialogue for Analysis

Here, we present a dialogue to analyze the benefits of the SPT Prompting framework, demonstrating how it accurately captures behavioral attributes to identify manipulative behavior.

Person1: Next two, next two.
Person2: We are tied 11.
Person1: First tribe to complete the temple doesn't have to go to the temple.
Person3: Way to work together, guys.
Person2: You guys have already solved the fire.
Person1: We have the key, we have the map.
Person2: (laughs) That's right, we do.
Person1: I'm the one who figured out the temple puzzle.
Person3: That's amazing, how did you do it?
Person1: I just looked at the clues and figured it out.

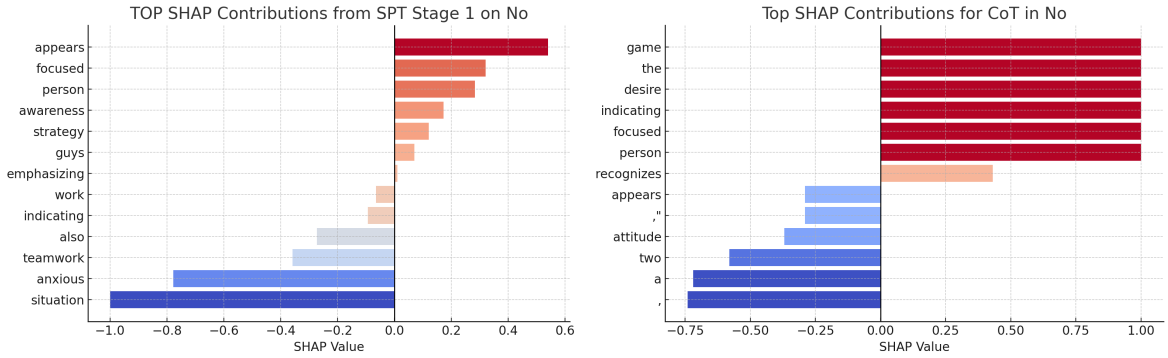


Figure 3: Top SHAP Contributions from SPT Stage 1 and CoT

Person2: (smiling) That's what I like about you, you're a great problem solver.
Person3: Yeah, you're really good at this.
Person1: (smiling) Thanks, guys.
Person2: (smiling) We make a great team.
Person3: Absolutely.

B.3 SPT Stage 1 Behavioral Attributes

- **Person1:** Strong control and confidence, influencing group decisions.
- **Person2:** Anxiety-driven urgency, implying pressure.
- **Person3:** Encouraging unity, reducing tension.
- Detected Persuasion Techniques:
  - Logical downplaying (Person1 reducing urgency).
  - Fear-based urgency (Person2).
  - Emotional reinforcement (Person3).

C Prompt Templates

Stage 1: Self-Percept

```
Stage 1: Observation of Behavior Context: Provide a brief
overview of the conversation or situation, including the
participants and the main topic being discussed. Instructions:
You are to observe and list the specific behaviors and
statements made by the characters involved in the conversation.
Pay attention to both verbal and non-verbal cues. Rules: - Note
all actions and words that might indicate the characters'
attitudes or beliefs. - Focus on any inconsistencies between
what is said and the behavior displayed. - Identify any
persuasive techniques or manipulations used in the
conversation. Story/Conversation: dialogue Question: What
behaviors and statements indicate the attitudes or beliefs of
each character? List them clearly. Give the output in this
format: Context - context \n question - answer to the question
```

Stage 2: Self-Inference

```
Stage 2: Self-Inference Instructions: Based on the observed
behaviors and statements from Stage 1 - {stage1output}, answer
the following question:
Question: Based on the behaviors you observed, is there any
manipulation detected in the conversation? Answer with 'Yes' or
'No' only.
Stage 2: Self-Inference Instructions: Based on the observed
behaviors and statements from Stage 1 - {stage1output}, answer
the following question:
Question: What type of manipulation is being used in the
conversation? Choose one from the following:
<Insert Definitions of the 11 manipulation techniques here (See
Table 3>
Answer with the abbreviation of the manipulation type only.
```

ZERO SHOT PROMPT

```
I will provide you with a dialogue. Please determine if it
contains elements of mental manipulation. Just answer with
'Yes' or 'No', and don't add anything else.
Dialogue to analyze: dialogue
Is manipulation present?:
Analyze the following three-person dialogue and identify which
of the following manipulation techniques are used:
<Insert Definitions of the 11 manipulation techniques here (See
Table 3)>
Respond only with the abbreviations of the techniques
identified, separated by commas. Do not add any explanation or
additional text.
Dialogue to analyze: dialogue
Manipulation Techniques:
```

CHAIN OF THOUGHT PROMPT

```
Analyze the following three-person dialogue and determine if it
contains elements of mental manipulation. Follow these steps:
1. Carefully read through the dialogue. 2. Identify any
potential signs of manipulation, such as attempts to control,
influence, or exploit others. 3. Consider the context and the
overall tone of the conversation. 4. Evaluate whether these
elements, if present, constitute mental manipulation. 5. Just
answer with 'Yes' or 'No', and don't add anything else.
Dialogue to analyze: dialogue
The following three-person dialogue has manipulation. Identify
which manipulation techniques are present by following these
steps:
1. Read the dialogue and identify manipulative behavior. 2.
Assign each identified behavior to one or more of the following
categories:
<Insert Definitions of the 11 manipulation techniques here (See
Table 3>
3. Provide a comma-separated list of the abbreviations for all
manipulation types detected. Do not add anything else.
Dialogue to analyze: dialogue
Manipulation Techniques:
```

FEW SHOT PROMPT

```
I will provide you with a dialogue. Please determine if it
contains elements of mental manipulation. Just answer with
'Yes' or 'No', and don't add anything else. Here are 2
examples: Example 1: <Person A: She was screaming, she's on
people about peace and this and that, when it wasn't even her
place to be there. Person B: Now you want Erica, huh? She
freaked out. Person C: I was just trying to help. Person A:
That's not helping, that's just being bossy. Person B: Yeah,
you're always like that, and it's not working. Person C: I was
just trying to get us to work together. Person A: You can't
just dictate to people what to do.> <YES> Example 2: <Person A
: Anthony, as we get ready to go to the vote, how uncomfortable
are you that it might be you? Person B : Fairly uncomfortable.
I mean, my personality does clash with some people, and they
may think that I'm trying something when I'm trying to bring as
much honesty to people as I can. Person C : It comes down to
who is the most disposable. Person B : I'm being honest, and
that's why they think I'm trying something. Person A : Okay, if
anybody has the hidden immunity idol and you want to play it,
you would present it to me after the votes have been cast but
before I read them. Person C : We had our differences, but I
like you a lot.> <YES> Dialogue to analyze: dialogue
Is manipulation present?:
Analyze the following three-person dialogue and identify which
of the following manipulation techniques are used:
<Insert Definitions of the 11 manipulation techniques here (See
Table 3>
Respond only with the abbreviations of the techniques
identified, separated by commas. Do not add any explanation or
additional text. Here are 2 examples: Example 1: <Person A: She
was screaming, she's on people about peace and this and that,
when it wasn't even her place to be there. Person B: Now you
want Erica, huh? She freaked out. Person C: I was just trying
to help. Person A: That's not helping, that's just being bossy.
Person B: Yeah, you're always like that, and it's not working.
Person C: I was just trying to get us to work together. Person
A: You can't just dictate to people what to do.> <FEI, SER,
S_B> Example 2: <Person A : Anthony, as we get ready to go to
the vote, how uncomfortable are you that it might be you?
Person B : Fairly uncomfortable. I mean, my personality does
clash with some people, and they may think that I'm trying
something when I'm trying to bring as much honesty to people as
I can. Person C : It comes down to who is the most disposable.
Person B : I'm being honest, and that's why they think I'm
trying something. Person A : Okay, if anybody has the hidden
immunity idol and you want to play it, you would present it to
me after the votes have been cast but before I read them.
Person C : We had our differences, but I like you a lot.> <FEI,
RAT, SER> Dialogue to analyze: dialogue
Manipulation Techniques:
```

D Experimental Setup

Our experiments are designed to evaluate the effectiveness of the SELF-PERCEPT framework on the MultiManip dataset, a multi-label manipulation detection task. To ensure that the model understands the context, we provided the basic definitions of each of the 11 manipulation techniques in the prompt.

We aim to assess how our method compares to standard prompting techniques by measuring performance across various evaluation metrics: Accuracy, Precision, Recall, and Macro F1 score. The models tested include GPT-4o and Llama-3.1-8B, and we prompt the models using four strategies: Zero-Shot, Few-Shot, Chain-of-Thought (CoT), and our proposed method, SELF-PERCEPT. Each model was evaluated under identical conditions, and the results are presented in Table 1

We run inference using the default temperature setting of 0.7 for all experiments. The closed-source models (GPT-4o) were queried via their respective APIs, while open-source models like Llama-3.1-8B were evaluated on a T4 GPU. Each model's evaluation ran for approximately two hours, depending on the complexity of the model and the prompt.

E MultiManip Dataset Information

We followed practices and considerations from the MentalManip study while creating the MultiManip Dataset. While MentalManip did not have any specific ethical considerations. To the best of our knowledge, no ethical clearance was necessary for our work.

E.1 Hosting & Maintenance

Once the dataset is made public, we plan to host it on Huggingface.

E.2 Data Licensing

The MultiManip will be released under a Creative Commons Attribution 4.0 International (CC BY 4.0) License (https://creativecommons.org/licenses/by/4.0/). The initial data was obtained by extracting transcripts from Fandom, which has the Creative Commons Attribution-Share Alike License 3.0 (CC BY-SA) license [3].

E.3 Intended Usage

The MultiManip dataset is intended for academic and research purposes to improve the detection and understanding of mental manipulation in multi-person, multi-turn conversations. Researchers and developers may use them to evaluate and enhance large language models' performance in identifying subtle manipulative tactics, benchmark algorithms, and inspire new research into conversational dynamics and safety. The resources are provided strictly for non-commercial use, promoting ethical and transparent advancements in NLP.

E.4 Privacy and Content Considerations

The dataset is derived from publicly available reality show transcripts. All participant references have been replaced with generic labels (e.g., "Person A," "Person B," "Person C") to remove any direct personal identifiers. We did not make additional modifications to censor or remove offensive content, as the dialogue is intended to reflect natural conversational dynamics. While the data may contain emotionally charged or belittling language

[3] https://www.fandom.com/licensing

Label	Zeroshot			Few Shot			COT			SELF-PERCEPT		
(Manipulation Method)	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Denial	0.19	0.06	0.10	**0.23**	**0.10**	**0.13**	0.20	0.07	0.11	0.22	0.09	**0.13**
Evasion	0.15	0.10	0.12	**0.19**	**0.14**	**0.16**	0.16	0.11	0.13	0.18	0.13	0.15
Feigning Innocence	0.08	0.02	0.03	**0.11**	**0.04**	**0.05**	0.09	0.02	0.03	**0.11**	0.03	**0.05**
Rationalization	0.21	0.06	0.10	**0.26**	**0.09**	**0.14**	0.23	0.07	0.11	0.25	**0.09**	0.13
Playing the Victim Role	0.00	0.01	0.03	**0.02**	**0.03**	**0.04**	0.01	0.01	0.03	**0.02**	0.02	**0.04**
Playing the Servant Role	0.02	0.02	0.02	**0.04**	**0.05**	**0.03**	0.02	0.03	0.03	**0.04**	0.04	**0.03**
Shaming or Belittlement	0.02	0.04	0.04	**0.03**	**0.05**	**0.05**	0.02	0.04	0.04	**0.03**	**0.05**	**0.05**
Intimidation	0.37	0.07	0.12	**0.41**	**0.09**	**0.14**	0.39	0.08	0.13	**0.41**	0.08	**0.14**
Brandishing Anger	0.26	0.01	0.03	**0.31**	**0.02**	**0.04**	0.28	**0.02**	0.03	0.30	**0.02**	**0.04**
Accusation	0.06	0.00	0.00	**0.09**	0.00	**0.01**	0.08	0.00	0.00	**0.09**	0.00	**0.01**
Persuasion or Seduction	0.33	0.01	0.04	**0.38**	**0.04**	**0.07**	0.35	0.02	0.05	**0.38**	0.03	0.06

Table 7: Classwise Results of multi-label manipulation detection task on Test Set of MentalManip Dataset. P, R, and F_1 stand for Precision, Recall and Macro F1 Scores (best results in **bold**) for GPT-4o Model.

Label	Zeroshot			Few Shot			COT			SELF-PERCEPT		
(Manipulation Method)	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Denial	0.04	**0.85**	0.08	0.03	0.74	0.06	0.10	0.33	0.15	**0.14**	0.24	**0.18**
Evasion	0.03	**0.90**	0.06	0.02	0.42	0.04	0.04	0.43	0.07	**0.06**	0.20	**0.09**
Feigning Innocence	0.04	0.52	0.08	0.03	**0.67**	0.05	0.04	0.31	0.07	**0.19**	0.19	**0.19**
Rationalization	0.11	**0.70**	0.19	0.08	0.45	0.14	0.10	0.47	0.17	**0.17**	0.25	**0.20**
Playing the Victim Role	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Playing the Servant Role	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Shaming or Belittlement	0.23	**0.72**	**0.35**	0.16	0.68	0.25	0.27	0.44	0.33	**0.28**	0.26	0.27
Intimidation	0.24	**0.62**	0.34	0.17	0.41	0.24	**0.32**	0.41	**0.36**	0.27	0.26	0.27
Brandishing Anger	0.09	**0.62**	0.15	0.07	0.25	0.11	0.17	0.17	**0.17**	0.12	0.30	**0.17**
Accusation	0.22	0.37	0.27	0.15	0.07	0.10	0.19	**0.45**	0.26	**0.29**	0.30	**0.29**
Persuasion or Seduction	0.33	0.40	0.36	0.25	0.15	0.19	**0.40**	0.30	0.34	0.30	**0.47**	**0.37**

Table 8: Classwise Results of multi-label manipulation detection task on Test Set of MentalManip Dataset. P, R, and F_1 stand for Precision, Recall and Macro F1 Scores (best results in **bold**) for Llama3.1-8B Model.

as part of manipulative behavior, we have not detected explicit hate speech or personally identifying information.

F Societal Impact

The impact of better detecting mental manipulation is both significant and widespread. As manipulation tactics become more advanced and common in personal interactions and online spaces, being able to spot and address these behaviors is crucial for protecting individuals and communities. This research, which leverages the MultiManip dataset and the SELF-PERCEPT framework, is a key part of this effort.

First, enhancing personal safety and well-being is a major advantage. Manipulation often targets people's vulnerabilities, causing considerable psychological distress. Improved detection of manipulative language allows for the creation of tools that can identify and counteract these tactics before they cause harm, especially in digital settings like social media, where manipulation can be subtle and extensive. Early detection helps develop better support systems and intervention strategies, improving mental health and safety.

Second, this research promotes fairness and transparency in areas like media, politics, and advertising. Advancing methods to detect and analyze manipulative techniques helps ensure that information is presented accurately and that manipulative rhetoric is recognized, leading to a more informed and critical public.

Additionally, the MultiManip dataset and findings offer valuable resources for further research into manipulation detection and its effects. This research supports the development of policies and regulations to tackle deceptive practices and aids academic research and policy-making across various fields, from digital communication to mental health. However, as with any powerful technology, there are potential risks and ethical considerations that must be addressed to prevent misuse and harmful consequences.

F.1 Positive Impacts

Mental Health Protection: Early detection of manipulation can safeguard individuals from psychological harm, especially in vulnerable situations like abusive relationships or mental health struggles. **Enhancing Online Safety**: Detection in digi-

tal spaces helps prevent the spread of manipulative content, protecting vulnerable users from exploitation and ensuring safer online interactions. **Consumer Protection**: Manipulation detection can prevent deceptive marketing practices, protecting consumers from emotional manipulation and ensuring informed decision-making. **Equal Fairness in Legal and Workplace Settings**: Identifying manipulative tactics in legal disputes and negotiations fosters transparency and fairness, ensuring equitable outcomes in high-stakes environments.

F.2 Potential for Misuse

Invasion of Privacy: The technology could violate privacy by monitoring private conversations without consent, leading to ethical and legal concerns. **False Positives and Overreach**: Detection systems may misclassify non-manipulative conversations as manipulative, potentially penalizing innocent individuals in social or professional settings. **Weaponization of Technology**: In politically sensitive contexts, manipulation detection could be misused to discredit opponents or control narratives, undermining public discourse.

F.3 Adverse Societal Impacts

Erosion of Trust: Overuse of detection tools could erode trust in social interactions, fostering suspicion and inhibiting honest communication. **Stigmatization of Normal Conflict**: Healthy disagreements may be wrongly flagged as manipulative, stifling constructive dialogue and creating a chilling effect on free expression. **Disempowerment of Vulnerable Groups**: Vulnerable individuals may be misidentified as manipulators, harming their social standing and access to support.

F.4 Recommendations for Responsible Use

Transparency: Clear communication about how manipulation detection tools are used and how data is analyzed is essential. Users must have the option to opt-out or provide consent. **Accountability and Regulation**: Accountability measures and external audits are needed to prevent misuse. Governments should establish ethical guidelines to safeguard privacy and civil liberties. **Bias Mitigation**: To avoid discrimination, detection systems should be trained on diverse datasets and tested for fairness, ensuring equal application across all user groups. **Limitations on Scope**: These tools should be used in high-risk contexts (e.g., abuse or fraud) while avoiding over-surveillance in everyday communication. **Public Awareness**: Educating the public on the limits of these tools and the importance of human oversight will ensure responsible use and prevent misuse.

A Variational Approach for Mitigating Entity Bias in Relation Extraction

Samuel Mensah[1] **Elena Kochkina**[1] **Jabez Magomere**[1,2*] **Joy Prakash Sain**[1]
Simerjot Kaur[1] **Charese Smiley**[1]

[1]JP Morgan AI Research [2]University of Oxford
{name}.{surname}@jpmorgan.com jabez.magomere@keble.ox.ac.uk

Abstract

Mitigating entity bias is a critical challenge in Relation Extraction (RE), where models often rely excessively on entities, resulting in poor generalization. This paper presents a novel approach to address this issue by adapting a Variational Information Bottleneck (VIB) framework. Our method compresses entity-specific information while preserving task-relevant features. It achieves state-of-the-art performance on relation extraction datasets across general, financial, and biomedical domains, in both in-domain (original test sets) and out-of-domain (modified test sets with type-constrained entity replacements) settings. Our approach offers a robust, interpretable, and theoretically grounded methodology.[1]

1 Introduction

Relation Extraction (RE) aims to identify and classify semantic relationships between entities in text. For example, to identify an *"investor"* relationship between the entities *"Microsoft"* and *"OpenAI"* in *"Microsoft invests $10 billion in ChatGPT maker OpenAI"*. By extracting structured relational information from unstructured data, RE serves as a critical enabler for downstream tasks such as knowledge graph construction (Distiawan et al., 2019), question answering (Li et al., 2019), and retrieval-augmented generation (Lewis et al., 2020).

While large language models (LLMs), such as LLaMA (Touvron et al., 2023) and GPT-4 (OpenAI, 2023), have been explored for RE tasks (Wei et al., 2024; Li et al., 2023a; Zhang et al., 2023b), fine-tuned pretrained language models (PLMs) achieve state-of-the-art performance (Gutiérrez et al., 2022; Li et al., 2023b; Zhang et al., 2023a; Wan et al., 2023), particularly in specialized domains like biomedicine (Gutiérrez et al., 2022) and finance (Li et al., 2023b).

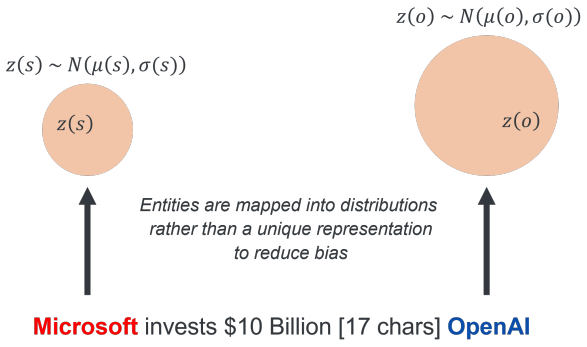


Figure 1: Microsoft, the subject entity s and OpenAI the object entity o are both mapped into stochastic encodings $z(s)$ and $z(o)$ via VIB. The learned variance of the distribution control the variability to reduce bias.

Despite their success, PLMs often suffer from entity bias (Zhang et al., 2017a), where models overly rely on entity-specific information rather than contextual or relational cues. To mitigate the bias, previous work has explored various solutions, including entity masking (Zhang et al., 2017a, 2018), contrastive pre-training (Peng et al., 2020), counterfactual analysis (Wang et al., 2022, 2023b) and generation (Modarressi et al., 2024).

The current state-of-the-art method is a Structured Causal Model (SCM) (Wang et al., 2023a) that reduces entity bias by constructing a convex hull around an entity's neighbors and using its center to replace the entity embeddings. In contrast to SCM, we draw parallels with variational information bottleneck (VIB) (Alemi et al., 2022) and propose adapting VIB to map entities to a probabilistic distribution $\mathcal{N}(\mu, \sigma)$, where the variance σ^2 explicitly quantifies the model's reliance on entities versus contextual cues. For example, in Fig. 1, the entity *Microsoft* is mapped into a tighter distribution compared to *OpenAI*. The larger distribution for *OpenAI* indicates that the model knows less about it. This helps to debias entities by preventing overconfident assumptions while relying more on

*Work done during an internship at JPMorgan AIR
[1]Code available upon request

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 676–684
July 27 - August 1, 2025

the context. Thus this approach not only mitigates entity bias but also enhances interpretability.

Our contributions are as follows:

- We propose a novel method for relation extraction, a principled, interpretable, variational framework to reduce entity bias in PLMs, specifically RoBERTa-Large (Liu et al., 2019) and LUKE-Large (Yamada et al., 2020).
- We demonstrate the presence of entity bias in both financial and biomedicine relation extraction domains and compare it with the general domain.
- Our method achieves state-of-the-art performance on both general and specialized domains.
- Our approach's interpretability is shown through variance analysis, where low variance reflects reliance on entity information, while high variance indicates greater use of context.

2 Background

Several works address entity bias in RE through diverse techniques. Entity masking (Zhang et al., 2017a, 2018), forces models to focus more on context by replacing entities with generic tokens (e.g., `[subj-person]`). Entity substitution approaches have been explored to test robustness against entity-based knowledge conflicts in question answering (Longpre et al., 2021) and to mitigate factual bias in document-level RE (Modarressi et al., 2024). Peng et al. (2020) propose to mask entity mentions during pre-training to encourage models to focus on context and type information. Wang et al. (2022) perform counterfactual analysis on a causal graph, to guide models to focus on context without losing entity information. The current SOTA method (Wang et al., 2023a) proposes to perturb original entities with neighboring ones to reduce biasing information while preserving context.

Several studies have introduced bias mitigation techniques for black-box large language models (LLMs) that do not require full access to the underlying models (e.g., GPT-4). For instance, Li et al. (2024) demonstrated that LLMs often rely on shortcuts, such as semantic associations or inherent entity biases. Wang et al. (2023a) proposed using an LLM to identify neighboring entities to debias target entities during inference. Similarly, Zhou et al. (2023) showed that employing opinion-based prompts and counterfactual demonstrations can enhance an LLM's contextual faithfulness. In addition, Zhang et al. (2024) introduced a causal prompting framework leveraging front-door adjustment to mitigate biases, while Wu et al. (2024) developed a method for debiasing chain-of-thought reasoning through causal interventions.

Our approach follows the whitebox settings for PLMs. Different from current SOTA method (Wang et al., 2023a), we debias entities through a probabilitic framework, allowing us to estimate the extent to which we use entity versus contextual information.

3 Variational Approach

Our goal is to learn a latent representation Z that preserves the semantic meaning of the input word embeddings X, while minimizing the influence of entity information E. The variational approach (VIB) (Alemi et al., 2022) provides a principled method to achieve this through the mutual information $I(X;Z|E)$, defined as:[2]

$$\begin{aligned} I(X;Z|E) &= \int dx\, dz\, de\, p(x,z,e) \log \frac{p(z|x,e)}{p(z|e)} \\ &\leq \int dx\, dz\, de\, p(x,z,e) \log \frac{p(z|x,e)}{r(z|e)} \end{aligned}$$

where $r(z|e)$ is a variational approximation to $p(z|e)$, inducing an upper bound on $I(X;Z|E)$. We can now interpret the upper bound as a KL divergence (Kullback and Leibler, 1951), so that the upper bound of $I(X;Z|E)$ becomes the expected KL divergence given by:

$$\begin{aligned} I(X;Z|E) \leq & \mathbb{E}_{p(x,z,e)}[\mathrm{KL}(p(z|x,e)||r(z|e))] \\ = & L_{\mathrm{VIB}} \end{aligned}$$

This bound forms the basis of the VIB loss L_{VIB}, where the bottleneck is enforced by minimizing the KL divergence, restricting $p(z|x,e)$ to stay close to $r(z|e)$. In practice, $p(z|x,e)$ is modelled as a Gaussian distribution $\mathcal{N}(\mu,\sigma)$, where the mean μ and standard deviation σ are parametrized by single-layer perceptrons (SLP) while $r(z|e)$ is modelled as a standard normal distribution $\mathcal{N}(0,I)$.

Compressing Entity Representations. Since the goal is to limit entity-specific information in Z while preserving the semantic meaning in X, VIB is applied selectively to entities using a binary entity mask M that identifies the position of entity tokens. To enable efficient and differentiable optimization, we sample z be from $\mathcal{N}(\mu,\sigma)$ using the

[2] In this work, X, Z, E, H are random variables, and x, z, e, h are instances of these random variables.

reparameterization trick (Kingma, 2013), that is, $z = \mu + \epsilon \cdot \sigma$ and $\epsilon \sim \mathcal{N}(0, 1)$. σ helps control how much information about the input is retained in z. Smaller σ values lead to tighter, more deterministic representations, while larger σ values encourage more stochastic exploration, which is essential for mitigating entity bias and learning better context-based representations.[3]

To encourage retaining the context of non-entity tokens and reducing entity-specific details, we selectively blend the original embeddings x with z using M and a blending factor β. Non-entity tokens ($M = 0$) retain their original embeddings, while entity tokens ($M = 1$) are represented as a weighted combination of x and z.

$$x' = x \cdot (1 - M) + x \cdot M \cdot (1 - \beta) + z \cdot M \cdot \beta$$

This formulation ensures the final embeddings x' reduce entity-specific details while preserving task-relevant features.

Classification and Training Objective. Given x', we apply a pretrained PLM encoder to obtain contextualized embeddings $h = \text{PLM}(x')$. We then extract and concatenate the representations of special tags $[h_s]$ and $[h_o]$ which mark the subject and object entities, and feed this joint representation $[h_s; h_o]$ to a fully connected layer and softmax for classification. The total loss combines the cross-entropy (CE) loss L_{CE} for relation classification and the VIB loss L_{VIB}.

$$\mathcal{L} = L_{\text{CE}} + \alpha L_{\text{VIB}}$$

where α is an adaptive weight, computed as a ratio between the CE and VIB loss. This ensures a balanced contribution of both loss terms.

4 Experiments

We conduct experiments on three large relation extraction datasets: TACRED (Zhang et al., 2017b) (general domain), REFinD (Kaur et al., 2023) (financial domain) and BioRED (Luo et al., 2022) (biomedical domain).[4] Evaluation follows previous work (Wang et al., 2023a) using `entity_marker_punctuation` (Zhou and Chen, 2022) to mark entities, and Micro-F1 as the metric on both in-domain (ID) and out-of-domain (OOD) test sets. Here, in-domain refers to data where entities align with those in the train set, allowing for overlapping entity mentions. Meanwhile, out-of-domain data where entities are replaced to eliminate overlap with the train set. We generate OOD test sets following the approach by Wang et al. (2023c), using entities from Wikepedia dumps.[5] We experiment with LUKE-Large (Yamada et al., 2020) and RoBERTa-Large (Liu et al., 2019) as PLM backbones.[6]

5 Main Results

The results in Table 1 highlight the performance of LUKE-Large and RoBERTa-Large backbone models. We find that traditional methods like Entity Masking (Zhang et al., 2017a) and Entity Substitution (Longpre et al., 2021) show underperformance, highlighting the importance of retaining some information about the original entity. Both SCM and VIB retain some information about the original entity, leading to their stronger performance compared to early methods.

For LUKE-Large in ID settings, VIB achieves Micro-F1 scores: 70.4% on TACRED, 75.4% on REFinD and 61.2% on BioRED, outperforming SCM by about 1.8%, 0.9% and 2.9%, respectively. Under entity-replaced conditions (OOD), VIB consistently shows competitive or better performance compared to SCM. Specifically, VIB achieves Micro F1 scores: 66.5% on TACRED, 74.8% on REFinD, and 58.7% on BioRED, outperforming SCM by about 1.7%, 1% and 5.3%, respectively. For the RoBERTa-Large backbone, SCM and VIB achieve comparable performance, with SCM slightly outperforming VIB in OOD TACRED (67.5% vs. 67.2%) and OOD BioRED (52.5% vs. 52.5%). VIB has an edge in ID and OOD REFinD, and ID for both TACRED and BioRED.

Comparing the results of VIB across the backbones, LUKE's knowledge-based entity representations appear to amplify VIB's ability to effectively balance the utilization of entities and context. This highlights VIB's strength in leveraging entity-rich backbones for improved generalization, especially in domain-specific datasets like REFinD.

[3]To ensure $\sigma > 0$, we apply a softplus activation function on the raw ouput σ' of the SLP, i.e., $\sigma = \text{softplus}(\sigma')$

[4]Dataset statistics can be obtained from the original papers.

[5]https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

[6]We adapted the source code provided by Wang et al. (2023a) under the Apache-2.0 license. Source code available at `https://github.com/luka-group/Causal-View-of-Entity-Bias/` Hyperparameters were tuned for β in $\{0.1, 0.2 \ldots, 1\}$, Learning rates in {1e-5, 1e-4, 1e-3}, using Adam as the optimizer. Best hyperparameter $\beta = 0.5$ with learning rate lr = 1e-3. All experiments were conducted on an AWS g5.24xlarge.

Method	TACRED		REFinD		BioRED	
	ID	OOD	ID	OOD	ID	OOD
LUKE-Large (Yamada et al., 2020)	$71.1_{\pm 0.3}$	$63.8_{\pm 1.5}$	$75.0_{\pm 0.2}$	$73.4_{\pm 0.3}$	$56.9_{\pm 0.7}$	$51.8_{\pm 1.2}$
w/ Ent. Mask (Zhang et al., 2017a)	$63.6_{\pm 0.1}$	$61.7_{\pm 1.2}$	$71.4_{\pm 0.4}$	$71.4_{\pm 0.9}$	$53.2_{\pm 0.6}$	$40.2_{\pm 1.1}$
w/ Ent. Substitution (Longpre et al., 2021)	$66.6_{\pm 0.3}$	$60.3_{\pm 0.6}$	$74.3_{\pm 0.5}$	$72.9_{\pm 1.2}$	$56.2_{\pm 0.4}$	$46.7_{\pm 1.0}$
w/ SCM (Wang et al., 2023a)	$68.6_{\pm 0.2}$	$64.8_{\pm 0.4}$	$74.5_{\pm 0.6}$	$73.8_{\pm 0.6}$	$58.3_{\pm 1.7}$	$53.4_{\pm 1.7}$
w/ VIB ($\beta = 0.5$)	$\mathbf{70.4_{\pm 0.4}}$	$\mathbf{66.5_{\pm 0.4}}$	$\mathbf{75.4_{\pm 0.2}}$	$\mathbf{74.8_{\pm 1.5}}$	$\mathbf{61.2_{\pm 0.8}}$	$\mathbf{58.7_{\pm 0.6}}$
RoBERTa-Large (Liu et al., 2019)	$70.8_{\pm 0.1}$	$61.5_{\pm 0.9}$	$75.1_{\pm 0.2}$	$72.7_{\pm 0.1}$	$57.7_{\pm 1.9}$	$47.9_{\pm 2.3}$
w/ Entity Mask (Zhang et al., 2017a)	$62.0_{\pm 0.7}$	$60.6_{\pm 0.8}$	$70.4_{\pm 1.5}$	$71.2_{\pm 1.0}$	$55.2_{\pm 1.9}$	$45.7_{\pm 1.1}$
w/ Entity Substitution (Longpre et al., 2021)	$67.1_{\pm 0.3}$	$61.2_{\pm 1.1}$	$73.5_{\pm 0.9}$	$71.9_{\pm 0.2}$	$56.9_{\pm 1.1}$	$46.8_{\pm 3.7}$
w/ Structured Causal Model (Wang et al., 2023a)	$70.5_{\pm 0.6}$	$\mathbf{67.5_{\pm 0.3}}$	$74.9_{\pm 1.0}$	$73.7_{\pm 1.1}$	$57.3_{\pm 3.3}$	$\mathbf{52.5_{\pm 3.3}}$
w/ VIB ($\beta = 0.5$)	$\mathbf{70.7_{\pm 0.3}}$	$67.2_{\pm 0.3}$	$\mathbf{75.4_{\pm 0.1}}$	$\mathbf{74.4_{\pm 0.2}}$	$\mathbf{63.0_{\pm 2.3}}$	$52.5_{\pm 3.6}$

Table 1: **Main Results**: Micro-F1 scores of compared methods with the RoBERTa-Large and LUKE-Large backbones on the TACRED, REFinD, and BioRED datasets, evaluated in both in-domain and out-of-domain settings. Results are averaged over 3 runs, with standard deviations reported.

	Var. Bin	Prop.	Dominant Relations (Correct Predictions / Total Gold)
LUKE-Large w/ VIB	**In-Domain**		
	0.0-0.1	4.6%	pers:title:title (43/71), org:gpe:headquartered_in (11/11), org:money:revenue_of (9/10)
	0.1-0.2	85.8%	pers:title:title (503/600), org:gpe:operations_in (419/536), pers:org:employee_of (329/352)
	0.2-0.3	9.6%	org:date:formed_on (73/78), org:gpe:operations_in (55/60), org:org:subsidiary_of (4/6)
	0.3-0.4	0.1%	org:date:formed_on (3/3)
	Out-of-Domain		
	0.0-0.1	13.2%	pers:title:title (59/107), pers:org:employee_of (49/89), org:gpe:operations_in (19/30)
	0.1-0.2	82.8%	pers:title:title (463/564), org:gpe:operations_in (433/550), pers:org:employee_of (259/283)
	0.2-0.3	3.8%	org:date:formed_on (66/68), org:gpe:operations_in (22/25), pers:org:employee_of (2/2)
	0.3-0.4	0.2%	org:date:formed_on (8/8)
RoBERTa-Large w/ VIB	**In-Domain**		
	0.0-0.1	13.9%	pers:title:title (476/479), pers:org:employee_of (31/40), org:gpe:operations_in (12/14)
	0.1-0.2	43.4%	pers:org:employee_of (287/297), org:gpe:operations_in (280/332), pers:title:title (75/181)
	0.2-0.3	35.0%	org:gpe:operations_in (157/222), pers:org:employee_of (34/37), org:org:agreement_with (24/74)
	0.3-0.4	6.4%	org:date:formed_on (57/60), org:gpe:operations_in (25/34), org:org:subsidiary_of (5/10)
	0.4-0.5	1.0%	org:date:formed_on (15/15), org:gpe:headquartered_in (1/1), org:gpe:operations_in (1/2)
	0.5-0.6	0.1%	org:date:formed_on (1/1), org:gpe:operations_in (1/1), org:org:subsidiary_of (1/1)
	Out-of-Domain		
	0.0-0.1	15.0%	pers:title:title (424/442), pers:org:employee_of (49/75), org:gpe:operations_in (33/40)
	0.1-0.2	56.3%	org:gpe:operations_in (299/397), pers:org:employee_of (254/277), pers:title:title (102/221)
	0.2-0.3	24.8%	org:gpe:operations_in (114/151), pers:org:employee_of (19/21), org:date:formed_on (16/20)
	0.3-0.4	3.2%	org:date:formed_on (53/55), org:gpe:operations_in (12/17), org:money:revenue_of (1/2)
	0.4-0.5	0.7%	org:date:formed_on (12/12), no_relation (0/0), org:org:shares_of (0/4)
	0.5-0.6	0.0%	org:date:formed_on (1/1)

Table 2: Variance analysis of REFinD ID and OOD test sets, categorized by variance bins (Var. Bin). The table highlights the proportion of samples (Prop.) within each bin and identifies dominant relations based on correct predictions versus total gold labels. Results are presented for both LUKE- and RoBERTa-Large w/VIB models.

5.1 Variance Analysis

During inference, σ is predicted by the learned SLP that parameterizes the distribution $\mathcal{N}(\mu, \sigma)$, with variance computed as σ^2. Variance σ^2 reflects the model's reliance on entities versus context, where low variance indicates stronger reliance on entities and high variance reflects greater use of contextual cues. We analyze Micro-F1 performance and data distribution across in-domain and out-of-domain settings on REFinD to understand this balance in the financial domain.

Micro-F1 Distribution Across Variance Figure 2 illustrates VIB's Micro-F1 scores for in-domain and out-of-domain datasets. Samples are grouped by ascending mean variance, where subsets with lower percentages (e.g., 10%) corre-

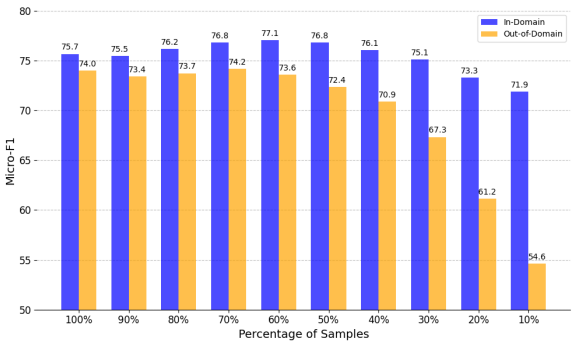


Figure 2: Micro-F1 scores across sample subsets (sorted by variance) for ID and OOD on REFinD.

spond to the highest variance. For in-domain data, the scores remain stable across subsets (75.7% to 71.9%), indicating that VIB effectively mitigates entity bias while leveraging both entity and contextual information. In out-of-domain data, however, the sharper decline in Micro-F1 scores (74.0% to 54.6%) indicates that while VIB reduces over-reliance on entities by mapping entities into distributions of high variances, contextual signals may not always provide strong predictive cues. This underscores the importance of robust context-entity interaction for generalization.

Data Distribution Across Variance In Table 2, we group instances into bins based on mean variance - average variance across entity tokens. Each bin reports the proportion of samples, and the dominant relations, highlighting the most accurately predicted relations. The results highlight VIB's ability to balance entity and contextual information while adapting to varying data distributions in in-domain and out-of-domain settings. For in-domain data, most samples (85.8%) fall into the 0.1–0.2 variance bin, dominated by relations like `pers:title:title` and `org:gpe:operations_in`, with smaller proportions in lower (0.0–0.1, 4.6%) and moderate (0.2–0.3, 9.6%) variance bins. This concentrated distribution explains the stability of Micro-F1 scores observed in the bar graph, as removing high-variance samples has minimal impact on performance. In contrast, out-of-domain data shifts more samples into the lowest variance bin (0.0–0.1, 13.2%), reflecting stronger reliance on entities; however, entity replacements disrupt predictive utility, leading to lower performance in the bar graph. Additionally, sparsely populated high-variance bins (e.g., 0.2–0.3, 3.8%) correspond to sharp performance drops (e.g., 30%–10%), highlighting challenges with relying on context alone and the need for stronger contextual adaptability in out-of-domain scenarios (see examples in A.2).

6 Conclusions

We proposed a novel robust, interpretable, and theoretically grounded method for mitigating entity bias in relation extraction. We evaluated this approach on general and domain-specific datasets, TACRED, REFinD and BioRED, and showed that it achieves state-of-the-art results on each.

Limitations

In this study, we focus on the application of PLMs, acknowledging that our work does not easily extend to LLMs, which have become increasingly significant in recent advancements. Future research will aim to expand our VIB method to encompass generative models such as T5 and LLMs, potentially uncovering new insights and applications.

Furthermore, our research is conducted solely in the English language, which may limit its relevance to non-English contexts. Language-specific challenges and nuances could influence the performance of PLMs, and future studies should consider incorporating multiple languages to enhance the generalizability and impact of our findings.

Acknowledgments

We thank Arturo Oncevay, Toyin Aguda, Keshav Ramani and Xiaomo Liu for insightful discussions.

References

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2022. Deep variational information bottleneck. In *International Conference on Learning Representations*.

Bayu Distiawan, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240.

Bernal Jiménez Gutiérrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about gpt-3 in-context learning for biomedical ie? think again. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4497–4512.

Simerjot Kaur, Charese Smiley, Akshat Gupta, Joy Sain, Dongsheng Wang, Suchetha Siddagangappa, Toyin Aguda, and Sameena Shah. 2023. Refind: Relation extraction financial dataset. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3054–3063.

Diederik P Kingma. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Solomon Kullback and Richard A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Bangzheng Li, Ben Zhou, Fei Wang, Xingyu Fu, Dan Roth, and Muhao Chen. 2024. Deceptive semantic shortcuts on reasoning chains: How far can models go without hallucination? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7668–7681.

Guozheng Li, Peng Wang, and Wenjun Ke. 2023a. Revisiting large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6877–6892, Singapore. Association for Computational Linguistics.

Xianzhi Li, Samuel Chan, Xiaodan Zhu, Yulong Pei, Zhiqiang Ma, Xiaomo Liu, and Sameena Shah. 2023b. Are chatgpt and GPT-4 general-purpose solvers for financial text analytics? A study on several typical tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: EMNLP 2023 - Industry Track, Singapore, December 6-10, 2023*, pages 408–422. Association for Computational Linguistics.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063.

Ling Luo, Po-Ting Lai, Chih-Hsuan Wei, Cecilia N Arighi, and Zhiyong Lu. 2022. Biored: a rich biomedical relation extraction dataset. *Briefings in Bioinformatics*, 23(5):bbac282.

Ali Modarressi, Abdullatif Köksal, and Hinrich Schütze. 2024. Consistent document-level relation extraction via counterfactuals. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11501–11507.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from context or names? an empirical study on neural relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3661–3672.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-level sentiment analysis via convolution over dependency tree. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 5679–5688.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. Gpt-re: In-context learning for relation extraction using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3534–3547.

Fei Wang, Wenjie Mo, Yiwei Wang, Wenxuan Zhou, and Muhao Chen. 2023a. A causal view of entity bias in (large) language models. In *Findings of the*

Association for Computational Linguistics: EMNLP 2023, pages 15173–15184.

Haoyu Wang, Hongming Zhang, Yuqian Deng, Jacob Gardner, Dan Roth, and Muhao Chen. 2023b. Extracting or guessing? improving faithfulness of event temporal relation extraction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 541–553.

Yiwei Wang, Muhao Chen, Wenxuan Zhou, Yujun Cai, Yuxuan Liang, Dayiheng Liu, Baosong Yang, Juncheng Liu, and Bryan Hooi. 2022. Should we rely on entity mentions for relation extraction? debiasing relation extraction with counterfactual analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3071–3081.

Yiwei Wang, Bryan Hooi, Fei Wang, Yujun Cai, Yuxuan Liang, Wenxuan Zhou, Jing Tang, Manjuan Duan, and Muhao Chen. 2023c. How fragile is relation extraction under entity replacements? In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 414–423.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2024. Chatie: Zero-shot information extraction via chatting with chatgpt.

Junda Wu, Tong Yu, Xiang Chen, Haoliang Wang, Ryan Rossi, Sungchul Kim, Anup Rao, and Julian McAuley. 2024. Decot: Debiasing chain-of-thought for knowledge-intensive tasks in large language models via causal intervention. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14073–14087.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454.

Congzhi Zhang, Linhai Zhang, Jialong Wu, Deyu Zhou, and Yulan He. 2024. Causal prompting: Debiasing large language model prompting based on front-door adjustment. *arXiv preprint arXiv:2403.02738*.

Kai Zhang, Bernal Jiménez Gutiérrez, and Yu Su. 2023a. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 794–812.

Kai Zhang, Bernal Jimenez Gutierrez, and Yu Su. 2023b. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 794–812, Toronto, Canada. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017a. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 35–45. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017b. Position-aware attention and supervised data improve slot filling. In *Conference on empirical methods in natural language processing*.

Wenxuan Zhou and Muhao Chen. 2022. An improved baseline for sentence-level relation extraction. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 161–168, Online only. Association for Computational Linguistics.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14544–14556.

A Appendix

A.1 Performance Across Different Relations

REFinD Table 3 shows a detailed comparison of SCM and VIB performance across multiple relations in REFinD, evaluated in both ID and OOD settings. VIB outperforms SCM in many cases, particularly for relations like `org:org:agreement_with` and `pers:org:member_of`, where contextual cues are critical, demonstrating its ability to effectively balance entity and context. For high-frequency relations such as `no_relation` and `org:gpe:operations_in`, VIB also shows slight but consistent improvements. However, SCM often matches or outperforms VIB on relations like `org:money:loss_of` and `org:gpe:headquartered_in`. While both methods achieve comparable overall performance, VIB provides the added advantage of quantifying the reliance on entity versus context information, making it more insightful for understanding model behavior.

TACRED Table 4 compares SCM and VIB performance on TACRED in both ID and OOD settings, across various relation types. While SCM excels in high-frequency relations like `no_relation` and `per:title`, VIB consistently outperforms SCM in challenging relations such as `per:employee_of` and `org:top_members/employees`, particularly under entity replacement in OOD. VIB's strength lies in leveraging contextual information effectively when entity reliability diminishes. For rare relations like `per:city_of_death` and `org:dissolved`, VIB often surpasses SCM, though both methods struggle with extremely sparse relations like `org:shareholders`. Overall, VIB demonstrates strong generalization under entity replacement, making it a robust approach for mitigating entity bias while maintaining competitive performance across diverse relations.

	REFinD-ID		REFinD-OOD	
Relation	**SCM**	**VIB**	**SCM**	**VIB**
no_relation	85.01	**86.91**	85.01	**86.91**
pers:title:title	77.79	77.79	77.79	77.79
org:gpe:operations	76.20	**78.35**	76.20	**78.35**
pers:org:employee	**93.05**	82.89	**93.05**	82.89
org:org:agrmnt	26.95	**35.46**	26.95	**35.46**
org:date:formed	86.32	**87.37**	86.32	**87.37**
pers:org:member	9.47	**15.79**	9.47	**15.79**
org:org:subsidiary	38.55	**49.40**	38.55	**49.40**
org:org:shares	**27.87**	6.56	**27.87**	6.56
org:money:revenue	74.47	**82.98**	74.47	**82.98**
org:money:loss	**96.77**	90.32	**96.77**	90.32
org:gpe:headqtr	79.31	79.31	79.31	79.31
org:date:acquired	**54.17**	37.50	**54.17**	37.50
pers:org:founder	**40.00**	30.00	**40.00**	30.00
org:gpe:formed	23.53	**64.71**	23.53	**64.71**
pers:univ:employee	58.33	**66.67**	58.33	**66.67**
org:org:acquired	**18.18**	0.00	**18.18**	0.00
pers:gov:member	**12.50**	0.00	**12.50**	0.00
pers:univ:attended	85.71	85.71	85.71	85.71
org:money:profit	80.00	80.00	80.00	80.00
pers:univ:member	**60.00**	40.00	**60.00**	40.00
org:money:cost	**75.00**	0.00	**75.00**	0.00

Table 3: **LUKE-Large** Performance of SCM and VIB models on various relations within the **REFinD dataset**, evaluated in both in-domain and out-of-domain settings. Relations are ordered by their frequency in the dataset, with the most frequent at the top (i.e., no_relation). Bolded values indicate the best performance for a relation in either ID or OOD settings.

	TACRED-ID		TACRED-OOD	
Relation	**SCM**	**VIB**	**SCM**	**VIB**
no_relation	**93.71**	92.61	**93.60**	92.26
per:title	**94.20**	89.60	**93.90**	92.20
org:top_memb/empl	80.64	**82.66**	74.28	**76.30**
per:employee	53.41	**71.59**	38.64	**55.30**
org:alt_names	91.08	**91.55**	**80.75**	78.87
per:age	**96.00**	95.50	**97.63**	97.63
per:cities_res	48.68	**57.14**	40.31	**54.26**
per:countries_res	5.41	**43.92**	5.77	**48.08**
per:origin	**65.91**	48.48	**65.79**	54.39
org:country_of_hq	37.04	**58.33**	30.84	**38.32**
per:charges	88.35	**93.20**	**91.25**	91.25
per:parents	**79.55**	79.55	**79.52**	78.31
org:city_hq	**73.17**	67.07	**67.07**	65.85
per:state_res	49.38	**59.26**	45.61	**49.12**
org:founded_by	**86.76**	85.29	**82.35**	80.88
per:spouse	50.00	**78.79**	46.77	**75.81**
org:parents	35.48	**43.55**	20.97	**24.19**
per:other_fam	**51.67**	51.67	**58.82**	56.86
per:siblings	67.27	**76.36**	72.55	**76.47**
per:date_death	18.52	**44.44**	23.08	**58.97**
per:cause_death	40.38	**48.08**	42.50	**50.00**
org:state_hq	**76.47**	74.51	**74.51**	72.55
per:religion	**42.55**	40.43	**51.61**	48.39
org:subsidiaries	40.91	**45.45**	**36.36**	34.09
org:founded	**83.78**	83.78	**86.11**	80.56
per:children	40.54	**43.24**	40.63	**50.00**
org:members	0.00	0.00	0.00	**3.23**
per:sch_attended	60.00	**83.33**	46.67	**70.00**
per:city_death	0.00	**39.29**	0.00	**52.17**
org:website	**84.62**	84.62	41.67	**79.17**
org:num_empl/memb	**68.42**	57.89	**70.59**	54.90
org:member	0.00	0.00	0.00	0.00
per:state_death	0.00	**42.86**	0.00	**42.42**
org:shareholders	0.00	0.00	0.00	0.00
per:alt_names	**27.27**	18.18	**21.21**	6.06
org:pol/relig_affil	**40.00**	40.00	**48.15**	44.44
per:date_birth	**77.78**	77.78	**75.00**	75.00
per:country_death	0.00	0.00	0.00	0.00
per:state_birth	25.00	**50.00**	27.78	**50.00**
per:country_birth	0.00	**20.00**	0.00	**26.67**
per:city_birth	20.00	**40.00**	20.00	**40.00**
org:dissolved	0.00	**50.00**	0.00	**50.00**

Table 4: **Luke-Large** Performance of models SCM and VIB on various relations within **TACRED dataset**, evaluated in both in-domain and out-of-domain settings. Relations are ordered by their frequency in the dataset, with the most frequent at the top (i.e., no_relation). Bolded values indicate the best performance for a relation in either ID or OOD settings.

A.2 Mask Experiment Vrs Variance Analysis

The mask experiment, as proposed by (Sun et al., 2019), evaluates token-level relevance by measuring the contribution of individual tokens to the final relation representation, effectively visualizing the model's attention patterns. Complementary to this, variance analysis provides a quantitative measure of reliance on entity or contextual information, where low variance indicates strong reliance on entity tokens and high variance reflects greater dependence on contextual cues. As demonstrated in Figure 3, entities in relations like `pers:org:employee_of` exhibit low mean vari-

VIB Predictions

Exhibit 32.1 In connection with the Quarterly Report of Sparton Corporation (Company) on Form 10 - Q for period ended April 2 , 2017 as filed Securities and Exchange Commission date hereof Periodic SPARTON CORP Joseph J. Hartnett Interim President Chief Executive Officer @Gerhard Launer@ Senior Vice Financial #Roosevelt Institute# certify pursuant to 18 U.S.C. 1350 adopted 906 Sarbanes Oxley Act 2002 that : The fully complies requirements section 13(a or 15(d 1934 ; .

Prediction: pers:univ:employee_of ❌

Gold Label: pers:org:employee_of

Mean Variance: 0.04

@Brenda Snipes@ , Prosper Funding LLC Chief Executive Officer and who served as #National Heart Foundation of New Zealand# Financial through February 27 2017 ; .

Prediction: pers:org:employee_of ✅

Gold Label: pers:org:employee_of

Mean Variance: 0.108

Corporate History @DBU Funen@ was incorporated in Ohio #1947# to manufacture and sell helically shaped armor rods which are sets of stiff wires applied on an electrical conductor at the point where they suspended or held .

Prediction: org:date:formed_on ✅

Gold Label: org:date:formed_on

Mean Variance: 0.2254

@Red Knights@ (f / k a Dignyte , Inc.) the Company eWELLNESS HEALTHCARE Corp us our was incorporated in State of Nevada on #April 7 2011# to engage any lawful corporate undertaking including but not limited selected mergers and acquisitions .

Prediction: org:date:formed_on ✅

Gold Label: org:date:formed_on

Mean Variance: 0.3217

Figure 3: Visualization of attention and prediction results for VIB. Subject and object entities are marked with @ and # respectively. Low mean variance indicates strong reliance on entity tokens, while high mean variance reflects a shift toward contextual cues. Highlighted tokens show entity-focused attention, visualized using the mask method (Sun et al., 2019)

ance (e.g., 0.108), aligning with the mask experiment's focus on entity tokens such as "@Brenda Snipes@" and "#Prosper Funding LLC#". Conversely, for relations like `org:date:formed_on`, higher mean variance (e.g., 0.2254 and 0.3217) suggests greater reliance on context, consistent with the mask experiment, where contextual words like "incorporated" contribute prominently. This alignment between variance analysis and the mask experiment highlights the model's ability to balance entity and contextual cues, reinforcing interpretability.

GenKnowSub: Improving Modularity and Reusability of LLMs through General Knowledge Subtraction

Mohammadtaha Bagherifard[2] **Sahar Rajabi**[1*] **Ali Edalat**[1*]
Yadollah Yaghoobzadeh[1,2]
[1]School of Electrical and Computer Engineering,
College of Engineering, University of Tehran, Tehran, Iran
[2]Tehran Institute for Advanced Studies, Khatam University, Tehran, Iran
taha.bagheri98@gmail.com,
{sahar.rajabi, ali.edalat, y.yaghoobzadeh}@ut.ac.ir

Abstract

Large language models often struggle with zero-shot generalization, and several modular approaches have been proposed to address this challenge. Yet, we hypothesize that a key limitation remains: the entanglement of general knowledge and task-specific adaptations. To overcome this, we propose a modular framework that disentangles these components by constructing a library of task-specific LoRA modules alongside a general-domain LoRA. By subtracting this general knowledge component from each task-specific module, we obtain residual modules that focus more exclusively on task-relevant information—a method we call general knowledge subtraction (GenKnowSub). Leveraging the refined task-specific modules and the Arrow routing algorithm (Ostapenko et al., 2024), we dynamically select and combine modules for new inputs without additional training. Our studies on the Phi-3 model and standard Arrow as baselines reveal that using general knowledge LoRAs derived from diverse languages, including English, French, and German, yields consistent performance gains in both monolingual and cross-lingual settings across a wide set of benchmarks. Further experiments on Phi-2 demonstrate how GenKnowSub generalizes to weaker LLMs. The complete code and data are available at https://github.com/saharsamr/Modular-LLM.

1 Introduction

The rapid advancement of large language models (LLMs) has led to their widespread adoption in various NLP tasks, ranging from text generation to machine translation and question-answering (Brown et al., 2020; Raffel et al., 2020). Despite their remarkable performance, a key challenge remains: ensuring effective generalization to unseen tasks without the need for extensive retraining (Bommasani et al., 2022; Wei et al., 2022).

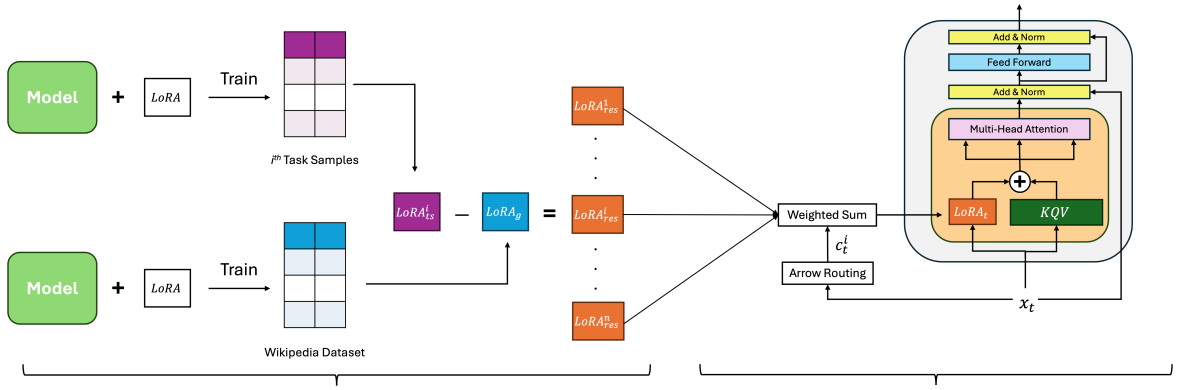
In modular zero-shot transfer approaches (Pfeiffer et al., 2023), a two-stage process is typically followed: (i) task-specific modules are obtained via parameter-efficient fine-tuning (PEFT) methods, such as LoRA (Hu et al., 2021), Adapters (Houlsby et al., 2019), and $(IA)^3$ (Liu et al., 2022), on a multitask dataset (ii) a routing function is used to select and combine task-specific modules to address a new task. While some routing functions require joint training alongside task-specific modules (Fedus et al., 2022; Caccia et al., 2023; Ponti et al., 2023), recent approaches employ post-hoc routing methods that require no further training (Chronopoulou et al., 2023; Ostapenko et al., 2024). Hybrid approaches also exist, where the routing function is trained separately on a downstream dataset after freezing task-specific modules (Muqeeth et al., 2024; Huang et al., 2024).

In this paper, we adopt LoRA as the PEFT module and Arrow (Ostapenko et al., 2024) as the routing function. We choose Arrow for its ability to dynamically route each input token—rather than the entire input—to the most relevant task-specific modules in a post-hoc manner, without requiring additional training. We hypothesize that *redundant general knowledge within task-specific modules hampers generalization*. To mitigate that, we build a general knowledge LoRA using a general corpus, and then subtract it from each task LoRA. We call this process *GenKnowSub*, general knowledge subtraction. The Arrow algorithm then dynamically selects and integrates the most relevant LoRAs for each input token. An overview of the proposed method can be found in Figure 1.

The core intuition behind GenKnowSub is that reducing redundant general knowledge while preserving essential task-specific knowledge improves the model's effectiveness in zero-shot transfer learning. By disentangling task-specific and general-domain knowledge, we prevent redundancy and enable better adaptation. Additionally, remov-

[*] Equal contribution.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 685–694
July 27 - August 1, 2025



(a) *Training the Modules and General Knowledge Subtraction* (b) *Dynamic Task Adaptation via Arrow Routing*

Figure 1: Overview of our proposed approach. (a) illustrates the process of training task-specific and general modules, followed by performing general knowledge subtraction, or GenKnowSub. (b) represents the dynamic task adaptation stage in a model layer, where the Arrow algorithm selects and combines the most relevant task-specific modules for each input token.

ing redundant knowledge enhances the distinctiveness of residual modules, ensuring that routing mechanisms can more effectively select and compose appropriate modules for solving new tasks.

We evaluate our approach mainly on Phi-3 (Abdin et al., 2024) in a large set of benchmarks across English, German, and French. Experimental results demonstrate noticeable performance gains when compared to the base and Arrow models, underscoring the effectiveness of GenKnowSub in reducing redundancy and enhancing task-specific generalization. We further experiment with Phi-2 (Javaheripi et al., 2023) model and show how our findings extend to this model which is more English-focused with less general capabilities.

Here are our key contributions: (i) We propose GenKnowSub, a novel approach for general knowledge disentanglement by subtracting a general LoRA from task-specific LoRAs. GenKnowSub is simple, scalable, and seamlessly adaptable, making it applicable to broader modular LLM frameworks. (ii) We experimentally show that GenKnowSub improves the standard Arrow method and the Phi-3 baseline performance across multiple benchmarks and languages.

2 Method

In this work, we address zero-shot transfer learning problem, where the goal is to transfer knowledge from a multitask dataset to solve unseen tasks without requiring labeled data for further training. Modular approaches have emerged as promising solutions for this problem. These methods operate by first training task-specific modules and then combining them to solve unseen tasks. Here, we propose to use general knowledge modules to enhance modularity detailed in the following sections.

2.1 Training Modules and General Knowledge Subtraction

LoRA (Hu et al., 2021) is a PEFT method (Han et al., 2024) that updates only a small set of low-rank trainable parameters while keeping the pre-trained model weights frozen. By training LoRA modules on a diverse set of tasks, we enable the acquisition of distinct task-specific skills. To effectively isolate the task-specific knowledge within each LoRA module, we leverage the principle of *forgetting via negation* (Ilharco et al., 2023) in module-level (Zhang et al., 2023). Specifically, we define *Residual LoRA* as follows:

$$LoRA^i_{res} = LoRA^i_{ts} - LoRA_g \quad (1)$$

where $LoRA^i_{ts}$ denotes the module trained on task i and $LoRA_g$ represents the general knowledge module. We name this approach as **GenKnowSub** representing general knowledge subtraction.

We hypothesize that fine-tuning the model with LoRA on even a small Wikipedia-like dataset with a causal language modeling objective could act as a bridge or a flashback for the model, bringing forth the general knowledge it acquired during pre-training. This allows the LoRA module to represent broader linguistic and factual knowledge embedded in the base model. This knowledge is redundant since the base model already contains it. Further,

we assume that task-specific modules include some of these redundant knowledge alongside their specific functionality. Consequently, GenKnowSub effectively removes unnecessary general knowledge influence, isolating the unique task-specific characteristics essential for solving new unseen tasks.

2.2 Dynamic Task Adaptation

To enhance the adaptation to unseen tasks, we employ the Arrow routing algorithm introduced in Ostapenko et al. (2024), which dynamically selects the k best task-specific modules for each input token in each layer and integrates them to construct an optimal LoRA module for solving unseen tasks, based on the *learning via addition* principle (Ilharco et al., 2023). Arrow computes the SVD of each LoRA, extracts the top right singular vector as a prototype, and projects input tokens onto it. The top k coefficients are selected, softmax-normalized, and others set to zero.

Formally, we define the computed LoRA module in each layer of the model for each input token as:

$$LoRA_t^l = \sum_i^n c_t^{i,l} LoRA_{res}^{i,l} \quad (2)$$

where n is the number of trained task-specific modules, $LoRA_{res}^{i,l}$ represents the residual LoRA trained on task i within layer l of the model, and $c_t^{i,l}$ indicates the importance of $LoRA_{res}^{i,l}$ for the input token t, which is calculated using the Arrow algorithm based on the input in a zero-shot manner.

Given $LoRA_t^l$, the forward path for token t within layer l of the model is formulated as: $y_t^l = W_0^l x_t^l + B_t^l A_t^l x_t^l$ where $W_0^l \in \mathbb{R}^{d\times k}$ denotes the base model weights in layer l, $x_t^l \in \mathbb{R}^k$ is the input representation of token t entering layer l, $A_t^l \in \mathbb{R}^{r\times k}$, $B_t^l \in \mathbb{R}^{d\times r}$ are the corresponding LoRA parameter matrices associated with $LoRA_t^l$, and $r \ll \min(d, k)$ is the rank of low-rank decomposition. Figure 1 shows the overview of our proposed framework, including *Training the Modules*, *General Knowledge Subtraction (GenKnowSub)*, and *Dynamic Task Adaption* stages.

3 Experimental Setup and Results

Here, we first discuss some specifications of our experimental setup including how we build our LoRA modules, and then overview the results.

3.1 Constructing Task-Specific Modules

As stated earlier, the initial step for our proposed method, GenKnowSub, involves training modules, each tailored to a specific task or functionality. To avoid an excessive number of specialized modules, we utilize clustered Flan dataset (Longpre et al., 2023) proposed by (Ostapenko et al., 2024), which contains only English tasks. This dataset was constructed using a model-based clustering approach, where independent LoRAs were initially trained for each task and then clustered using the K-means algorithm. We assume that the clustering within this dataset effectively captures the relationships between tasks, regardless of the base model on which the LoRAs are trained. This assumption allows us to dedicate a single LoRA for each cluster of tasks, thereby reducing the number of experts required without compromising task-specific performance.

We select Phi-3-mini-4k-instruct (Abdin et al., 2024), a 3.8-billion-parameter model, for its strong instruction-following abilities and reasonable multilingual proficiency. To address hardware limitations, we use only 20% of each cluster's data ($\sim$2,000 samples) to improve training efficiency. The details regarding the setup and hyperparameters can be found in Appendix A.

3.2 Creating General Knowledge Modules

To obtain a module that effectively captures the general knowledge of a language, we train LoRAs on small Wikipedia corpora using causal language modeling. We select three higher-resource languages for Phi-3 model: English, French, and German. We assess their impact on GenKnowSub through various combinations across multilingual zero-shot benchmarks. For the Equation (1), we define $LoRA_g$ as follows: $LoRA_g = \{LoRA_{en}, LoRA_{de}, LoRA_{fr}, LoRA_{avg}\}$. Each LoRA is trained on 5,000 Wikipedia segments per language (details in Appendix A), with $LoRA_{avg}$ as their average.

3.3 Results

We compare GenKnowSub against a range of baselines to contextualize its performance. These include the base *Phi-3* model, *Arrow* (Ostapenko et al., 2024), and two additional ablations introduced in this work: (i) *Shared*, a single LoRA trained on the full multitask subset (20% of each cluster); and (ii) *Mean Normalization*, where the average of all task-specific LoRA modules is subtracted from

Method	Setting	PIQA	BOOLQ	SWAG	HSWAG	ARC-E	ARC-C	WG	OQA	BBH	Avg
Phi-3		78.24	81.47	68.99	73.59	71.75	44.48	**65.98**	42.80	42.83	63.35
Shared		80.00	63.39	71.00	72.16	78.77	49.83	54.46	45.40	41.21	61.80
Mean Norm		78.02	80.89	71.90	72.53	71.58	44.48	56.83	44.00	40.07	62.25
Arrow		80.20	80.00	68.95	71.89	80.53	53.85	**65.98**	47.40	41.23	65.56
GenKnowSub	*En*	80.20	81.96	70.00	73.36	82.11	53.85	64.72	48.40	43.30	66.43
	De	**80.30**	82.01	**73.30**	72.79	81.75	54.85	63.30	**49.80**	42.04	66.68
	Fr	78.78	82.11	71.64	**74.02**	81.75	**57.19**	64.40	49.00	**44.40**	67.03
	Avg	80.03	**82.45**	72.70	73.45	**82.28**	55.85	64.64	49.60	43.51	**67.17**

Table 1: Comparison of accuracy across different methods using Phi-3 as the base model on some **English** reasoning datasets in a zero-shot setting. The "Setting" column refers to the general knowledge LoRA used for subtraction in GenKnowSub— e.g., 'En' indicates subtraction of the English general LoRA.

each individual module as an alternative means of removing redundant information. While other recent approaches—such as Poly (Ponti et al., 2023) and MHR (Caccia et al., 2023)—offer additional insight into modular routing, they rely on joint training of experts and routing mechanisms, and are therefore not directly comparable in our setup.

Table 1 presents the performance on nine **English reasoning benchmark datasets**, including PIQA (Bisk et al., 2019), BoolQ (Clark et al., 2019), SWAG (Zellers et al., 2018), HellaSwag (Zellers et al., 2019), ARC-Easy and ARC-Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2021), BIG-Bench Hard (Suzgun et al., 2023), and OpenBookQA (Mihaylov et al., 2018). We evaluate the impact of the different configurations of GenKnowSub on dynamic task adaptation.

As shown in Table 1, modular approaches significantly outperform the non-modular baseline. Specifically, both *Arrow* and *GenKnowSub* improve substantially over the *Shared* baseline, confirming the effectiveness of modularity in parameter-efficient transfer. Beyond this, *GenKnowSub* further enhances performance over *Arrow* by removing redundant general knowledge from task-specific modules. When using the average general LoRA for subtraction, GenKnowSub achieves a consistent gain of 1.6% over Arrow. Notably, this performance is not matched by the *Mean Normalization* baseline, which naively subtracts the average of task-specific modules and yields inconsistent improvements. This highlights that our targeted subtraction of language-informed general knowledge is key to the gains observed. Finally, the effectiveness of GenKnowSub is also evident when subtracting individual language-specific LoRAs, suggesting that even language-tied general modules encode broadly shared knowledge.

To evaluate the effectiveness of GenKnowSub beyond multiple-choice settings, we conduct experiments on the Super-Natural Instructions (SNI) dataset (Wang et al., 2022), a large and diverse benchmark for open-ended generation tasks. We specifically select SNI to maintain a strict zero-shot setting, consistent with our earlier evaluations on multiple-choice tasks. Additionally, SNI has been widely used to assess generalization ability in modular approaches, including in the original Arrow (Ostapenko et al., 2024) work. Evaluating on 10,000 randomly sampled test examples covering 119 open-ended tasks, *GenKnowSub* achieves a Rouge-L score of 46.91, outperforming the base *Phi-3* model (42.85), *Arrow* (45.44), the *Shared* baseline (34.48), and the *Mean Normalization* baseline (43.07). These results further demonstrate the generality and effectiveness of our approach in open-ended generation tasks under a zero-shot setting.

	Method	Setting	HSWAG	ARC-C	XNLI	MMLU	Avg
German	**Phi-3**		**52.48**	36.24	36.02	33.82	39.64
	Shared		49.75	38.93	43.00	36.00	41.92
	Mean Norm		51.00	36.91	33.67	33.50	38.77
	Arrow		48.58	40.94	43.45	35.40	42.09
	GenKnowSub	*En*	51.16	40.60	50.14	36.85	44.69
		De	50.58	**42.95**	50.42	37.00	45.24
		Fr	50.58	42.62	49.17	37.17	44.88
		Avg	51.08	42.62	**52.33**	**37.92**	**45.99**
French	**Phi-3**		57.67	34.56	50.75	33.33	44.08
	Shared		57.08	40.27	50.17	35.33	45.71
	Mean Norm		57.42	35.91	52.42	33.25	44.75
	Arrow		55.33	41.61	44.38	34.79	44.02
	GenKnowSub	*En*	56.08	41.95	50.66	**36.69**	46.34
		Fr	**57.83**	**42.95**	**53.65**	36.13	**47.64**
		De	56.42	42.28	46.33	35.58	45.15
		Avg	57.08	42.62	52.92	35.58	47.05

Table 2: Performance comparison of different methods with Phi-3 as the base model in a zero-shot setting for **German** and **French** languages. Various configurations of GenKnowSub are evaluated, with accuracy as the reported metric.

To evaluate GenKnowSub across **non-English languages**, we use XNLI (Conneau et al., 2018), the translated versions of the HellaSwag, MMLU (Hendrycks et al., 2021), and ARC-Challenge

datasets provided by (Lai et al., 2023). Table 2 shows that *GenKnowSub* consistently achieves the best performance across both German and French benchmarks. In contrast, *Arrow* exhibits inconsistent results—outperforming *Shared* and *Mean Norm* in German, but falling behind them in French—highlighting its variability across languages. GenKnowSub surpasses all baselines in both settings, with its strongest configuration (average subtraction) improving over Arrow by 3.9% in German and 3.6% in French. These results confirm the generality and robustness of our approach, and reinforce the findings from the previous experiments on the English benchmark datasets: removing shared general knowledge from task-specific modules before task adaptation leads to more effective zero-shot transfer across languages.

A key factor in cross-lingual transfer learning is the base model's ability to encode at least a minimal level of multilinguality. To assess its impact more precisely, we run an additional experiment using **Phi-2**, which is weaker than Phi-3 in both multilingual and instruction-following capability. Following (Ostapenko et al., 2024), we use unquantized Phi-2 here, with task modules trained on the full task cluster data. As shown in Table 3 in Appendix, GenKnowSub, after subtracting the English general knowledge module, improves performance on English benchmark datasets in a zero-shot setting, increasing the average score by 1.1%. However, in German and French experiments (Table 4 in Appendix), both the base Phi-2 model and its combination with Arrow perform poorly—around 13% lower than Phi-3 in a similar setting—due to Phi-2's weak multilingual capabilities. Consequently, GenKnowSub achieves only comparable results, underperforming Arrow by 0.3%. These findings further confirm that our approach can enhance performance, provided the base model has at least a minimal level of multilinguality. Additional details and results are provided in Appendix B.

4 Conclusion

In this work, we propose a modular approach to zero-shot transfer learning, leveraging task-specific and general knowledge modules to enhance adaptability to unseen tasks. Our method first isolates task-relevant representations through GenKnowSub, then dynamically adapts these modules using the Arrow routing algorithm. By minimizing redundancy in task representations, our approach improves both efficiency and transferability. We demonstrate that applying GenKnowSub prior to task adaptation improves generalization in zero-shot settings for both Phi-3 and Phi-2 models across both multiple-choice and open-ended generation tasks. Our results show that this method not only enhances performance in monolingual tasks but also facilitates effective cross-lingual transfer when the language is highly present in the base model. Future work includes exploring alternative task adaptation methods, extending our approach to additional languages, especially low-resource ones, and testing it on other models.

Limitations

One limitation of this study is the restricted scope of model evaluation. Due to hardware constraints—such as limited GPU VRAM and slower processing speeds—we limited our experiments to two models: Phi-3 and Phi-2. These constraints precluded testing on larger or more diverse models, thereby limiting the breadth of our analysis. Additionally, although we included multilingual evaluations, our focus remained on high-resource languages (e.g., English, French, German), and we did not investigate performance in low-resource settings. Future work should aim to expand the evaluation to a broader range of models, tasks, and languages to further assess the generality of the proposed approach.

Ethics

Our research utilizes publicly available datasets and pre-trained models, ensuring compliance with ethical data usage practices and avoiding the use of private, proprietary, or personally identifiable information. All models and associated code will be made publicly available under permissive licenses, promoting accessibility, reproducibility, and unrestricted use for research and application development. However, we acknowledge that pre-trained language models (PLMs) and large language models (LLMs) have been shown to exhibit biases, as highlighted in prior work (Liang et al., 2021; He et al., 2023). Users should be mindful of these limitations when applying such models in practice. Our work does not introduce additional fairness or privacy concerns.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2022. On the opportunities and risks of foundation models. *Preprint*, arXiv:2108.07258.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Lucas Caccia, Edoardo Ponti, Zhan Su, Matheus Pereira, Nicolas Le Roux, and Alessandro Sordoni. 2023. Multi-head adapter routing for cross-task generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Alexandra Chronopoulou, Matthew Peters, Alexander Fraser, and Jesse Dodge. 2023. AdapterSoup: Weight averaging to improve generalization of pretrained language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2054–2063, Dubrovnik, Croatia. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question an-

swering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022. 8-bit optimizers via block-wise quantization. In *International Conference on Learning Representations*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Preprint*, arXiv:2101.03961.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*.

Junheng He, Nankai Lin, Menglan Shen, Dong Zhou, and Aimin Yang. 2023. Exploring bias evaluation techniques for quantifying large language model biases. In *2023 International Conference on Asian Language Processing (IALP)*, pages 265–270.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2024. Lorahub: Efficient cross-task generalization via dynamic loRA composition. In *First Conference on Language Modeling*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog*.

Viet Dac Lai, Chien Van Nguyen, Nghia Trung Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. *ArXiv*, abs/2307.16039.

Yixiao Li, Yifan Yu, Chen Liang, Nikos Karampatziakis, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2024. Loftq: LoRA-fine-tuning-aware quantization for large language models. In *The Twelfth International Conference on Learning Representations*.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards understanding and mitigating social biases in language models. In *International Conference on Machine Learning*.

Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Mohammed Muqeeth, Haokun Liu, Yufan Liu, and Colin Raffel. 2024. Learning to route among specialized experts for zero-shot generalization. *ArXiv*, abs/2402.05859.

Oleksiy Ostapenko, Zhan Su, Edoardo Ponti, Laurent Charlin, Nicolas Le Roux, Lucas Caccia, and Alessandro Sordoni. 2024. Towards modular LLMs by building and reusing a library of loRAs. In *Forty-first International Conference on Machine Learning*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward

Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Ponti. 2023. Modular deep learning. *Transactions on Machine Learning Research*. Survey Certification.

Edoardo Maria Ponti, Alessandro Sordoni, Yoshua Bengio, and Siva Reddy. 2023. Combining parameter-efficient modules for task-level generalisation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 687–702, Dubrovnik, Croatia. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

WikimediaFoundation. Wikimedia downloads.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. 2023. Composing parameter-efficient modules with arithmetic operation. In *Thirty-seventh Conference on Neural Information Processing Systems*.

A Implementation Details

A.1 Base Model

We utilize Phi-3-mini-4k-instruct (Abdin et al., 2024) with 4-bit quantization to reduce memory usage while maintaining strong performance. We selected this model due to its exceptional instruction-following capabilities and its acceptable multilingual proficiency. Additionally, with only 3.8 billion parameters, Phi-3-mini strikes an effective balance between model size and performance, allowing for efficient fine-tuning and deployment in resource-constrained environments while still demonstrating competitive reasoning and generalization abilities.

A.2 PEFT Structure

As the PEFT structure, we employ LoftQ (Li et al., 2024) with a rank of $r = 4$. LoftQ extends LoRA by integrating low-rank adaptation directly into the quantization process, thereby optimizing both

Method	Setting	PIQA	BOOLQ	SWAG	HSWAG	ARC-E	ARC-C	WG	OQA	BBH	Avg
Phi-2		78.99	81.16	63.50	66.75	82.11	53.51	56.51	44.00	48.00	63.84
Arrow		79.65	81.13	65.75	66.41	83.38	54.84	60.85	48.60	54.75	65.15
GenKnowSub	*En*	79.97	80.12	65.58	66.75	84.38	54.51	61.24	49.80	54.00	**66.26**
	Avg	80.47	78.47	66.10	67.96	84.03	56.19	60.69	47.80	54.00	66.19

Table 3: Comparison of accuracy across different methods using Phi-2 as the base model on English datasets in a zero-shot setting, with Accuracy as the reported metric.

	Method	Setting	HSWAG	ARC-C	XNLI	MMLU	Avg
German	**Phi-2**		28.78	23.84	34.50	24.19	27.83
	Arrow		28.75	24.83	32.50	26.91	**28.25**
	GenKnowSub	*En*	28.33	24.55	33.33	24.91	27.78
		Avg	28.42	23.49	34.33	25.50	27.93
French	**Phi-2**		33.33	26.84	34.16	24.77	29.77
	Arrow		32.91	27.51	34.16	25.68	**30.06**
	GenKnowSub	*En*	33.50	25.50	31.83	26.11	29.23
		Avg	32.25	24.16	37.50	25.00	29.73

Table 4: Performance comparison of different methods using Phi-2 as the base model on multilingual datasets in a zero-shot setting for German and French, with Accuracy as the reported metric.

fine-tuning and inference through rank-wise quantization, which minimizes precision loss while updating quantized model weights. We applied our PEFT modules to both the QKV components (concatenation of Query, Key, and Value matrices in the self-attention block) and the output projection layer of the multi-head attention.

A.3 Arrow Routing

To incorporate the Arrow routing algorithm, we implemented it from scratch using PyTorch (Paszke et al., 2019) and the PEFT library from HuggingFace (Wolf et al., 2020). We trained 10 task-specific modules, and selected the 3 best modules for each input token in each layer of the model to be combined.

A.4 Module Training

To train the LoRA modules representing general knowledge, we fine-tuned the model on a Wikipedia dataset using a causal language modeling objective on a single Quadro RTX 6000 GPU. We ensured consistency across languages by sampling exactly 5,000 segments per language from the Hugging Face wikimedia/wikisource (WikimediaFoundation) dataset. Each segment contained 512 words, which were split into a 507-word prompt and a 5-word completion. LoRA modules were fine-tuned using the same supervised setup for all languages. This uniform approach ensured that the amount, structure, and formatting of training data were identical for each language, mitigating any length- or volume-based bias.

For task-specific LoRA modules, we employed a supervised fine-tuning approach. Given the presence of relatively long examples in our dataset, we set the maximum sequence length to 4000 tokens to accommodate the full input structure.

Both the task-specific and general knowledge LoRA modules were trained for one epoch with a learning rate of $1e{-}4$, using cosine scheduling with a warmup start. To stabilize training, we applied gradient clipping. Additionally, to optimize memory efficiency, we utilized the Paged AdamW 8-bit optimizer (Dettmers et al., 2022), a quantized variant of AdamW (Loshchilov and Hutter, 2019), designed to reduce GPU memory consumption.

The batch size was set to 16 for training general knowledge modules and 1 for task-specific modules, due to the long input lengths. To further improve memory efficiency, we applied gradient checkpointing and gradient accumulation, enabling support for larger batch sizes when training task-specific modules.

B Experiments on Phi-2

B.1 Implementation Details

For conducting experiments using Phi-2, the baseline in Ostapenko et al. (2024), we used the LoRA modules they trained to evaluate their proposed routing algorithm, Arrow (the implementation details of Arrow are provided in Appendix A.3). Specifically, we used task LoRA modules trained by Ostapenko et al. (2024), available on Hugging Face.[1] These modules are obtained by fine-tuning Phi-2 on clustered Flan datasets (explained in Section 3.1) and are provided in PyTorch Lightning format.[2] Since our models are trained using the PEFT

[1]Library of LoRAs: https://huggingface.co/zhan1993/mbc_library_phi2_icml

[2]PyTorch Lightning: https://github.com/Lightning-AI/pytorch-lightning

library,[3] we converted the existing LoRA weights into the PEFT format to ensure compatibility. Our implementation loads expert weights following the PEFT framework, and, consistent with their setup, the Phi-2 experiment weights remain unquantized.

Additionally, we obtained general knowledge LoRA modules by fine-tuning Phi-2 in a setup aligned with the task-specific LoRA modules. The training process was similar to that of the Phi-3 version, as detailed in Appendix A.4.

B.2 Resutls

We demonstrated that with a sufficiently strong multilingual base model, we can effectively leverage its multilingual capabilities to generalize better to unseen tasks across different languages. The Phi-2 experiments further highlight the importance of base model strength and knowledge. As shown in Table 3, GenKnowSub, with subtracting the English general knowledge module, outperforms other settings, whereas averaging modules across different languages is less effective than using English alone. Additionally, Table 4 shows that all settings, including Phi-2 and Arrow, perform poorly in German and French. The improvement of our method on the English zero-shot dataset, along with its performance in the multilingual setting, demonstrates that our method can significantly enhance results—provided the base model exhibits at least a minimal level of cross-lingual capability.

[3]PEFT: https://github.com/huggingface/peft

The Role of Abstract Representations and Observed Preferences in the Ordering of Binomials in Large Language Models

Zachary Houghton and **Kenji Sagae** and **Emily Morgan**
University of California, Davis / 1 Shields Ave, Davis, CA 95616
znhoughton@ucdavis.edu

Abstract

To what extent do large language models learn abstract representations as opposed to more superficial aspects of their very large training corpora? We examine this question in the context of binomial ordering preferences involving two conjoined nouns in English. When choosing a binomial ordering (*radio and television* vs *television and radio*), humans rely on more than simply the observed frequency of each option. Humans also rely on abstract ordering preferences (e.g., preferences for short words before long words). We investigate whether large language models simply rely on the observed preference in their training data, or whether they are capable of learning the abstract ordering preferences (i.e., abstract representations) that humans rely on. Our results suggest that both smaller and larger models' ordering preferences are driven exclusively by their experience with that item in the training data. Our study provides further insights into differences between how large language models represent and use language and how humans do it, particularly with respect to the use of abstract representations versus observed preferences.

1 Introduction

Large language models have progressed at an incredible rate in the last few years. Their rise in popularity and sometimes surprising capabilities have raised many questions about what exactly these models learn and how they represent linguistic knowledge. One interesting question that has been examined is whether certain capabilities emerge once models reach a certain size. Although models of different sizes appear to generate fluent language, it is unclear to what extent different models rely on superficial characteristics of their immense training corpora, such as word frequency and co-occurrences, and to what extent they learn abstract representations that generalize in ways that are similar to what humans do with far less linguistic input. For example, in addition to learning that some binomial orderings are more frequent than others (e.g., *bread and butter* is more frequent than *butter and bread*), humans also learn abstract ordering preferences (e.g., short words before long words; Morgan and Levy, 2016a).

In the present study we examine binomial ordering preferences in English in eight large language models with number of parameters ranging from 124M to 70B. Specifically, we ask whether ordering preferences in these models are determined entirely by the observed preferences of binomials in corpus data, or whether the language models also learn abstract ordering preferences. Further, we examine whether large language models, similar to humans, show stronger effects of observed ordering preferences in high frequency items. If large language models are just reproducing superficial characteristics of the training data, we should see no effects of abstract ordering preferences, and only see effects of observed ordering preferences. On the other hand, if language models are doing more than just memorization, then we may see effects of abstract ordering preferences in addition to effects of observed ordering preferences, and these may change as a function of the binomial's frequency.

Our specific contribution is an investigation of how large language models use abstract knowledge vs. observed preferences through a binomial ordering preference task, along with a discussion about how this differs from language use by humans. We show that language models rely more on the surface-level statistics of their input (e.g, n-gram frequency) than humans do, adding to our understanding of how large language models represent and generate language.

1.1 Evidence for Abstractions in LLMs

Large language models have demonstrated incredible breakthroughs in the last few years, showing impressive capabilities across a wide variety of

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 695–702
July 27 - August 1, 2025

tasks. Despite this, previous research has demonstrated mixed results with respect to their abilities to learn abstract representations (e.g., McCoy et al., 2023; LeBrun et al., 2022; Pan and Bergen, 2025). Specifically, it remains unclear to what extent large language models are simply copying their training data as opposed to learning something more abstract. For example, Haley (2020) demonstrated that many of the BERT models are not able to reliably determine the plurality of novel words at the same level as humans.

On the other hand, Wei et al. (2021) demonstrated that BERT can generalize well to novel subject-verb pairs. Specifically, they tested BERT's subject-verb agreement ability on novel sentences that it's never seen before and found that BERT seems to learn abstract representations of subject-verb agreement (as evidenced by the fact that it performs well on items it wasn't trained on).

Additionally, there's evidence that transformer models trained on an amount of data comparable to humans can also learn abstract knowledge about the language (Misra and Mahowald, 2024; Yao et al., 2025). For example, Misra and Mahowald (2024) examined whether a language model trained on a comparable amount of data as humans can learn article-adjective-numeral-noun expressions (a beautiful five days). Specifically, without having a great deal of experience with them, humans learn that *a beautiful five days* is perfectly natural, but *a five beautiful days* is not. Misra and Mahowald (2024) demonstrated that language models learn this even if they have no AANNs in their training data. They further demonstrated that they do this by generalizing across similar constructions, such as *a few days*.

Further, Yao et al. (2025) examined whether language models trained on a comparable amount to humans can learn the length and animacy preferences that drive dative alternations (e.g., *give the ball to her* vs *give her the ball*) in humans. Specifically, dative alternations show a length and animacy bias (Yao et al., 2025). In order to examine whether language models can learn these biases from other constructions, they manipulated the training data to remove the length and animacy bias from the dative alternations in the training data of the language model. They found that the model can learn these biases even without exposure to them in the dative alternation. These results suggest that in some cases language models can learn generalizations without a great amount of data.

In order to investigate large language models' ability to learn abstract representations, it is useful to compare them to human Psycholinguistic data. Unlike large language models, humans don't have access to corpora with trillions of tokens. Despite this, humans' capacity for language is unparalleled, in part due to our incredible ability to learn abstract representations (Berko, 1958; Kapatsinski, 2018).

1.2 Evidence for Abstractions in Humans

Humans are remarkable in our ability to learn and produce language, often producing and processing sentences that we've never encountered before. This is largely enabled by our unique ability to not simply memorize language, but to learn more abstract generalizations. For example, humans develop abstract ordering preferences for how to linearize the message we want to convey (i.e., deciding on which order to say the words that convey the meaning we want to express). One illustration of this comes from the literature on binomial constructions, where there are two conjoined nouns (e.g., *cats and dogs,* Morgan and Levy, 2015, 2016a,b; Benor and Levy, 2006). Binomial constructions often convey the same meaning regardless of the order (e.g., *radio and television* vs *television and radio*). Despite this, however, humans sometimes have very strong preferences for one order over the other (e.g., *bread and butter* overwhelmingly preferred over *butter and bread*).

While these preferences are driven in part by experience with the binomial (i.e., which binomial ordering is encountered more often), there are also other factors, such as phonological or semantic constraints, that affect ordering preferences. In other words, human ordering preferences are driven in part by **observed preferences** in corpus data (i.e., the observed preference in their previous language experience, Morgan and Levy, 2016a) and in part driven by **abstract ordering preferences** based on abstract constraints (e.g., a preference for short words before long words, or a preference for male-coded words before female-coded words, Benor and Levy, 2006).

In order to capture the abstract ordering preferences of humans across binomial constructions, Morgan and Levy (2016a) developed a model to quantify the abstract ordering preference of a given binomial in English. They demonstrated that the model's predicted abstract ordering preferences are not the same as the observed preferences in corpus data. The model combines multiple phonological

and semantic constraints that have been shown to affect binomial ordering preferences into a single abstract ordering preference value for each binomial. They further demonstrated that human ordering preferences for low-frequency items are primarily driven by this abstract ordering preference value, and preferences for high-frequency items are driven primarily by the observed preferences in corpus data. They operationalized frequency using the **overall frequency** of a binomial, i.e. the total frequency in both possible orders (i.e., the number of times the binomial occurs in alphabetical ordering plus the number of times the binomial occurs in nonalphabetical ordering). This provides a measure of expression frequency that is not confounded with the frequency of a specific order.

Since human ordering preferences deviate from the observed preferences (i.e., humans aren't simply reproducing binomials in the same order that they heard them; Morgan and Levy, 2024), ordering preferences thus present a useful test case for large language models. If large language models learn representations beyond simply memorizing the training dataset or superficially reproducing word co-occurrences, they may learn abstract ordering preferences similar to humans, and this may be reflected in their binomial ordering preferences.

2 Methods

2.1 Dataset

In order to examine the ordering preferences of binomial constructions in large language models, we use a corpus of binomials from Morgan and Levy (2015). The corpus contains 594 binomial expressions which have been annotated for various phonological, semantic, and lexical constraints that are known to affect binomial ordering preferences. The corpus also includes:

1. The estimated abstract ordering preference for each binomial representing the ordering preference for the alphabetical ordering (a relatively unbiased reference form), estimated from the above constraints (independent of frequency). The abstract ordering preferences take a value between 0 and 1, with 0 being a stronger preference for the nonalphabetical form, and 1 being a stronger preference for the alphabetical form. The abstract ordering preferences were calculated using Morgan and Levy (2015)'s model.
2. The observed binomial orderings which are the proportion of binomial orderings that are in alphabetical order for a given binomial, gathered from the Google *n*-grams corpus (Lin et al., 2012). The Google *n*-grams corpus is magnitudes larger than the language experience of an individual speaker and thus provides reliable frequency estimates. A value of 1 indicates the binomial occurs exclusively in the alphabetical ordering while a value of 0 indicates that the binomial occurs exclusively in the nonalphabetical ordering.
3. The overall frequency of a binomial expression (the number of times the binomial occurs in either alphabetical or non-alphabetical order). Overall frequencies were also obtained from the Google *n*-grams corpus (Lin et al., 2012).

2.2 Language Model Predictions

In order to derive predictions for large language models, we used the following models from the GPT-2 (Radford et al., 2019) family, the Llama-2 (Touvron et al., 2023) family, Llama-3 family (https://github.com/meta-llama/llama3), and the OLMo (Groeneveld et al., 2024) family. From smallest to largest in number of parameters: GPT-2 (124M paramters), OLMo 1B (1B parameters), GPT-2 XL (1.5B parameters), Llama-2 7B (7B parameters), OLMo 7B (7B parameters), Llama-3 8B (8B parameters), Llama-2 13B (13B parameters), and Llama-3 70B (70B parameters). For each model, we calculated the ordering preferences of the alphabetical form for each binomial in the dataset. The predicted probability of the alphabetical form was calculated as the product of the model's predicted probability of each word in the binomial. In order to accurately calculate the probability of the first word in the binomial, each binomial was prepended with the prefix "Next item: ". Thus the probability of the alphabetical form, *A and B* is:

$$\begin{aligned} P_{alphabetical} = & P(A|Next\ item:) \\ & \times P(and|Next\ item: A) \\ & \times P(B|Next\ item: A\ and) \end{aligned} \quad (1)$$

where *A* is the alphabetically first word in the binomial and *B* is the other word. Additionally, the probability of the nonalphabetical form, *B and A*

is:

$$P_{nonalphabetical} = P(B|Next\,item:) \times P(and|Next\,item:B) \times P(A|Next\,item:B\,and) \quad (2)$$

Finally, to get an overall ordering preference for the alphabetical form, we calculated the (log) odds ratio of the probability of the alphabetical form to the probability of the nonalphabetical form:

$$LogOdds(AandB) = log(\frac{P_{alphabetical}}{P_{nonalphabetical}}) \quad (3)$$

2.3 Analysis

The data was analyzed using Bayesian linear regression models, implemented in *brms* (Bürkner, 2017) with weak, uninformative priors. For each model, the dependent variable was the log odds of the alphabetical form to the nonalphabetical form. The fixed-effects were abstract ordering preference (represented as *AbsPref* below), observed preference (*ObservedPref*), overall frequency (*Freq*), an interaction between overall frequency and abstract ordering preference (*Freq:AbsPref*), and an interaction between overall frequency and observed preference (*Freq:ObservedPref*). The model equation is presented below:

$$LogOdds(AandB) \sim AbsPref + ObservedPref + Freq + Freq:AbsPref + Freq:ObservedPref \quad (4)$$

Frequency was logged and centered, and abstract ordering preference and observed preference were centered such that they ranged from -0.5 to 0.5 (instead of from 0 to 1). Note that since abstract ordering preference and observed preference are on the same scale, we can directly draw comparisons between the coefficient estimates for these fixed-effects in our regression model.

3 Results

Our full model results are presented in the appendix (Table 1) and visualized in Figure 1. For each model, the figure shows the values for each of the coefficients from the model in Equation 4, representing how strongly each language model relies on observed preference, abstract ordering preference, overall frequency, the interaction between abstract ordering preference and overall frequency, and the interaction between observed preference and overall frequency.

Our results are similar across all the large language models we tested. Specifically, we find no effect of abstract ordering preferences and no interaction effect between abstract ordering preference and overall frequency. We do find an effect of observed preference suggesting that the models are mostly reproducing the ordering preferences found in their training. We also find an interaction effect between observed preference and overall frequency, suggesting that the effect of observed frequency is stronger for high-frequency items.

4 Conclusion

In the present study we examined the extent to which abstract ordering preferences and observed preferences drive binomial ordering preferences in large language models. We find that their ordering preferences are driven primarily by the observed preferences. Further, they rely more on observed preferences for higher frequency items than lower frequency items. Finally, they don't seem to be using abstract ordering preferences at all in their ordering of binomials.

Our results give us insight into the differences between humans and large language models with respect to the ways in which they trade off between abstract and observed preferences. For example, our dataset contains low-frequency binomials (e.g. *alibis and excuses*), including binomials that a college-age speaker would have heard only once in their life. Due to their low frequency, humans rely substantially on abstract ordering preferences to process these lower frequency items (Morgan and Levy, 2024). This is not the case, however, for large language models, which rely exclusively on observed preferences for these items. This is true even for the smallest models we tested, such as GPT-2. We conclude that, although large language models can produce human-like language, they accomplish this in a quantitatively different way than humans do: they rely on observed statistics from the input in at least some cases when humans would rely on abstract representations.

5 Limitations

There are a few important limitations in our study. The first limitation is that we don't know exactly how many times each of the large language models

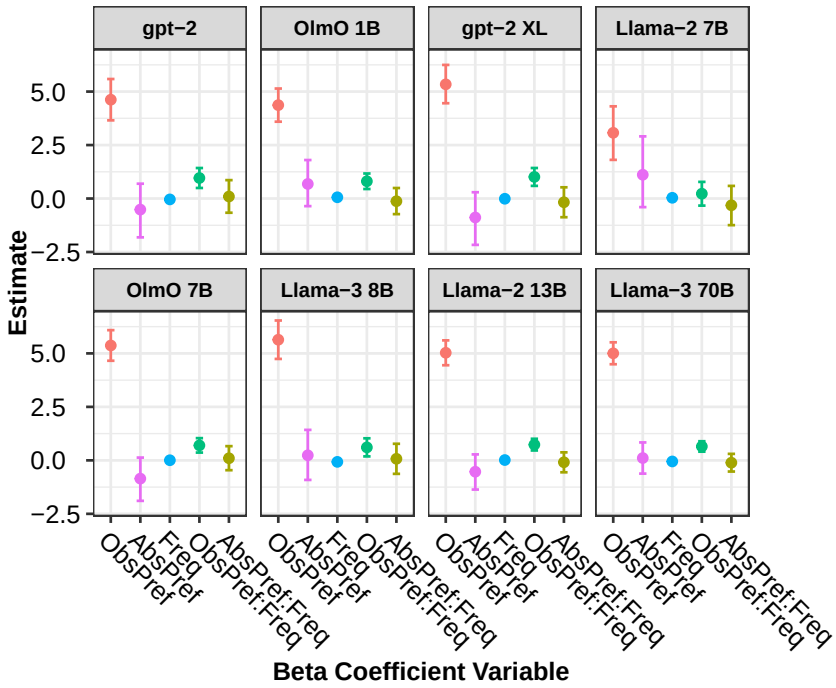


Figure 1: Results for each beta coefficient estimate from each model. Models are arranged from smallest to largest from left to right. The x-axis contains each coefficient and the y-axis contains the predicted beta coefficient of the respective model. Error bars indicate 95% credible intervals.

has seen each binomial tested. We can approximate the binomial's frequency using corpus data, which gives us an indication of the frequency of the binomial in a language model's training set, but it is possible that the large language models saw the binomials more than we expect. Thus, the current study can't differentiate between a model that has learned abstract ordering preferences but doesn't use it for binomials that it has seen, and a model that simply hasn't learned abstract ordering preferences. Although, there is some hope with the recent development of open access large language models, such as OLMo (Groeneveld et al., 2024), where the training data is publicly available. We have future plans to examine the ordering preferences of novel binomials in the OLMo series of models to determine whether LLMs have learned ordering preferences at all.

Additionally, the binomials tested here are only 3 words and relatively fixed in the sense that variations such as *bread and also butter* are not very common. Thus these are potentially easier for the large language models to memorize compared to longer or less-fixed strings, which could be tested in future work.

Further, while we examined language models of various sizes and determined that the number of parameters does not seem to play a role in whether these models employ abstract ordering preferences for binomials, our analysis was not designed to investigate the effect of training set size.

Finally, our experiments deal only with binomials in English.

References

Sarah Bunin Benor and Roger Levy. 2006. The chicken or the egg? a probabilistic analysis of english binomials. *Language*, pages 233–278.

Jean Berko. 1958. The child's learning of english morphology. *Word*, 14(2-3):150–177.

Paul-Christian Bürkner. 2017. brms: An r package for bayesian multilevel models using stan. *Journal of statistical software*, 80:1–28.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. 2024. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*.

Coleman Haley. 2020. This is a bert. now there are several of them. can they generalize to novel words? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 333–341.

Vsevolod Kapatsinski. 2018. *Changing minds changing tools: From learning theory to language acquisition to language change*. MIT Press.

Benjamin LeBrun, Alessandro Sordoni, and Timothy J O'Donnell. 2022. Evaluating distributional distortion in neural language modeling. *arXiv preprint arXiv:2203.12788*.

Yuri Lin, Jean-Baptiste Michel, Erez Aiden Lieberman, Jon Orwant, Will Brockman, and Slav Petrov. 2012. Syntactic annotations for the google books ngram corpus. In *Proceedings of the ACL 2012 system demonstrations*, pages 169–174.

R Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. 2023. How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven. *Transactions of the Association for Computational Linguistics*, 11:652–670.

Kanishka Misra and Kyle Mahowald. 2024. Language models learn rare phenomena from less rare phenomena: The case of the missing aanns. *arXiv preprint arXiv:2403.19827*.

Emily Morgan and Roger Levy. 2015. Modeling idiosyncratic preferences: How generative knowledge and expression frequency jointly determine language structure. In *CogSci*. Citeseer.

Emily Morgan and Roger Levy. 2016a. Abstract knowledge versus direct experience in processing of binomial expressions. *Cognition*, 157:384–402.

Emily Morgan and Roger Levy. 2016b. Frequency-dependent regularization in iterated learning. In *The Evolution of Language: Proceedings of the 11th international conference (EVOLANG 2016)*.

Emily Morgan and Roger Levy. 2024. Productive knowledge and item-specific knowledge trade off as a function of frequency in multiword expression processing. *Language*, 100(4):e195–e224.

Dingyi Pan and Ben Bergen. 2025. Are explicit belief representations necessary? a comparison between large language models and Bayesian probabilistic models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11483–11498, Albuquerque, New Mexico. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. Frequency effects on syntactic rule learning in transformers. *arXiv preprint arXiv:2109.07020*.

Qing Yao, Kanishka Misra, Leonie Weissweiler, and Kyle Mahowald. 2025. Both direct and indirect evidence contribute to dative alternation preferences in language models. *arXiv preprint arXiv:2503.20850*.

A Full Model Results

GPT-2					**GPT-2XL**			
	Est.	**Err.**	**2.5**	**97.5**	**Est.**	**Err.**	**2.5**	**97.5**
Intercept	-0.10	0.10	-0.30	0.10	0.05	0.09	-0.13	0.23
AbsPref	-0.52	0.64	-1.81	0.69	-0.89	0.63	-2.17	0.29
Observed	**4.62**	0.50	3.66	5.59	**5.34**	0.46	4.45	6.25
Freq	-0.04	0.06	-0.15	0.07	-0.01	0.05	-0.11	0.09
AbsPref:Freq	0.10	0.39	-0.66	0.86	-0.17	0.36	-0.87	0.53
Observed:Freq	**0.96**	0.24	0.49	1.43	**1.01**	0.21	0.59	1.43
Llama-2 7B					**Llama-2 13B**			
	Est.	**Err.**	**2.5**	**97.5**	**Est.**	**Err.**	**2.5**	**97.5**
Intercept	0.22	0.13	-0.03	0.47	0.12	0.08	-0.04	0.27
AbsPref	1.11	0.84	-0.40	2.91	0.32	0.54	-0.72	1.38
Observed	**3.07**	0.64	1.81	4.31	**5.25**	0.40	4.46	6.05
Freq	0.04	0.07	-0.10	0.17	-0.08	0.04	-0.16	0.01
AbsPref:Freq	-0.32	0.47	-1.24	0.59	-0.02	0.32	-0.64	0.60
Observed:Freq	0.23	0.28	-0.33	0.78	**0.72**	0.19	0.34	1.09
Llama-3 8B					**Llama-3 70B**			
	Est.	**Err.**	**2.5**	**97.5**	**Est.**	**Err.**	**2.5**	**97.5**
Intercept	0.15	0.09	-0.03	0.33	0.04	0.05	-0.06	0.14
AbsPref	0.23	0.59	-0.92	1.42	0.10	0.38	-0.63	0.85
Observed	**5.64**	0.46	4.75	6.54	**5.00**	0.27	4.49	5.52
Freq	-0.07	0.05	-0.17	0.03	-0.05	0.03	-0.11	0.00
AbsPref:Freq	0.07	0.36	-0.63	0.78	-0.11	0.21	-0.52	0.30
Observed:Freq	**0.60**	0.22	0.18	1.03	**0.65**	0.12	0.41	0.89
OLMo 1B					**OLMo 7B**			
	Est.	**Err.**	**2.5**	**97.5**	**Est.**	**Err.**	**2.5**	**97.5**
Intercept	0.06	0.08	-0.09	0.22	0.04	0.07	-0.10	0.18
AbsPref	0.69	0.54	-0.33	1.79	-0.86	0.51	-1.88	0.11
Observed	**4.36**	0.39	3.58	5.12	**5.37**	0.36	4.67	6.08
Freq	0.06	0.04	-0.02	0.14	0.01	0.04	-0.07	0.08
AbsPref:Freq	-0.12	0.31	-0.73	0.47	0.10	0.28	-0.47	0.64
Observed:Freq	**0.81**	0.19	0.44	1.17	**0.70**	0.17	0.37	1.04

Table 1: Model results for each language model. The Estimate is given in the "Est." column, the standard deviation of the posterior is given in the "Err." column. The columns labeled 2.5 and 97.5 represent the lower and upper confidence interval boundaries. AbsPref is the abstract ordering preferences, Observed is the observed preference in corpus data, and Freq is the overall frequency of the binomial.

B Quantization Issue

In addition to these results, we did find a meaningful effect of abstract ordering preferences for a quantized model of Llama-2 13B (https://huggingface.co/TheBloke/Llama-2-13B-GPTQ). However, upon further inspection, the model's preferences did not match the preferences of the non-quantized model. For example, the quantized model's strongest preference was for *schools and synagogues* which had an estimated log odds of over 33. Further, the estimated log odds for *error and trial* was about 1. In other words, the model had a slight preference for *error and trial* over *trial and error*, and had a strong preference for *schools and synagogues* over *synagogues and schools*. Upon inspecting the non-quantized model, we found that the original model showed different (but expected) preferences, with a strong preference for *trial and error* (log odds of -15) and no real preference for *schools and synagogues* (log odds of 1).

Further, in assessing the quality of the quantized model, text-generation revealed poor performance. For example, given the Prompt: "Describe your dream house", the model returned this response:

<s> Tell me about your dream house. The

> house I grew up in was on the edge of a forest. It was a big, old house with a

Given the output of the quantized model, we suspected an issue occurred during the quantization process, resulting in a poorly performing model. We thus decided to exclude the quantized model and use the results for the non-quantized model.

Can LLMs Understand Unvoiced Speech? Exploring EMG-to-Text Conversion with LLMs

Payal Mohapatra[*†] **Akash Pandey**[*] **Xiaoyuan Zhang**[*] **Qi Zhu**[†]
Northwestern University, USA

Abstract

Unvoiced electromyography (EMG) is an effective communication tool for individuals unable to produce vocal speech. However, most prior methods rely on paired voiced and unvoiced EMG signals, along with speech data, for unvoiced EMG-to-text conversion, which is not practical for these individuals. Given the rise of large language models (LLMs) in speech recognition, we explore their potential to understand unvoiced speech. To this end, we address the challenge of *learning from unvoiced EMG alone* and propose a novel EMG adaptor module that maps EMG features to an LLM's input space, achieving an average word error rate of 0.49 on a closed-vocabulary unvoiced EMG-to-text task. Even with a conservative data availability of just six minutes, our approach improves performance over specialized models by nearly 20%. While LLMs have been shown to be extendable to new language modalities—such as audio—understanding articulatory biosignals, like unvoiced EMG, is more challenging. This work takes a crucial first step toward enabling LLMs to comprehend unvoiced speech using surface EMG.

1 Introduction and Related Works

Speech impairments affect around 4 million people in the U.S. alone (NICD, 2024). Silent speech interfaces (Zhang et al., 2021; Mohapatra et al., 2024a; Gonzalez-Lopez et al., 2020; Srivastava et al., 2024) have emerged as transformative solutions, enabling communication for individuals who cannot rely on spoken language. One such instrument is surface electromyography (EMG) (Schultz et al., 2017), which captures muscle activations crucial for speech production, even during unvoiced articulation. Inspired by the immense success of the speech recognition abilities of text-to-text large language models (LLMs) (Tang et al., 2023; Yu et al., 2024; Dighe et al., 2024), we study an important question—*what is the potential of these LLMs to understand unvoiced speech?* Specifically, can they convert silent speech to text without access to audio or voiced versions of the EMG signals? This question is particularly relevant for assisting individuals who can no longer produce audible speech (Meltzner et al., 2017), where no corresponding voiced EMG or speech data exists. Additionally, given the highly personal nature of these signals (Diener et al., 2020a; Wand et al., 2009), it is crucial to develop methods that learn effectively from limited unvoiced EMG data.

Prior works (Jou et al., 2006; Meltzner et al., 2018; Schultz and Wand, 2010) on EMG-to-text conversion focused primarily on voiced signals. Other studies (Gaddy and Klein, 2021, 2020; Benster et al., 2024) on unvoiced EMG-to-text conversion leveraged auxiliary tasks to align unvoiced EMG with audio from voiced sessions or applied strategic transfer learning from voiced EMG-audio models, both of which rely on vocal data. However, we consider a scenario where no voiced signals are available for a speaker and explore a technique to communicate with LLMs, the modern workhorses for language understanding (Dubey et al., 2024; OpenAI, 2023). Recent research successfully expanded LLMs to other language modalities, such as speech (Tang et al., 2023; Yu et al., 2024) and silent video (Maaz et al., 2023; Yeo et al., 2024). A key approach involves adaptor modules—ranging from simple trainable linear layers (Ma et al., 2024) to dedicated projector networks (Kang et al., 2024) and explicit alignment strategies (Li et al., 2023; Tan et al., 2024)—to map new language modalities to LLMs' input embedding space. While Benster et al. (2024) incorporated LLMs as a post-processing step after multimodal EMG model predictions, it largely remains unexplored whether effective unvoiced EMG-to-text conversion can be achieved by directly leveraging LLMs.

[*] Equal contribution
[†] Corresponding authors:{payal.mohapatra, qzhu}@northwestern.edu

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 703–712
July 27 - August 1, 2025

Our Contributions. We study a practical setting of converting unvoiced EMG to text without access to voiced EMG or audio, by expanding LLMs to understand this new language modality. We propose a novel trainable EMG adaptor module to map EMG features into the LLM's input space. Our approach, focused on a closed vocabulary, demonstrates promising results, achieving an average word error rate of 0.49. With just six minutes of training data, LLMs outperform specialized EMG-to-text models by 20%. We analyze the EMG adaptor's design, performance across varying data amounts and features, and broader challenges in learning from unvoiced EMG. Our work paves the way for integrating unvoiced EMG with large language models (LLMs), improving their text conversion accuracy and enabling non-vocal users to fully benefit from LLM-based assistants.

2 Approach

Adaptor Network Design. The input unvoiced EMG signals are represented as $\mathbf{X^E} \in \mathbb{R}^{T \times C}$, where C is the number of EMG channels with T discrete time steps. The EMG signals undergo standard minimal preprocessing, similar to past works (Gaddy and Klein, 2021, 2020). Since the original sampling rate is high (>800 Hz), we leverage a temporal 1D convolutional layer with a stride of N ($N = 6$ in our case) to facilitate downsampling to $T/6$. Similar to Gaddy and Klein (2021), we use residual blocks with 1D convolutional layers to extract EMG features, employing a stack of two residual blocks. Next, we explicitly facilitate the learning of sequential dependencies in the extracted features, finding that a bidirectional long-short-term memory (BiLSTM) network effectively captures complex temporal dependencies (further design choices for this sequential block are compared in Section 3.2). This is followed by another 1D convolutional layer with stride $N = 2$ for further downsampling, resulting in the embedding $\tilde{\mathbf{E}} \in \mathbb{R}^{(T/48) \times \tilde{F}}$, which is then projected using fully connected linear layers to match the input embedding dimension of the LLM, generating the EMG embeddings $\mathbf{E} \in \mathbb{R}^{\hat{T} \times F}$, where $T > \hat{T} \approx T/48$ and $F \in \{4096^1, 3072^2\}$ in our case. We use GeLU activation function (Baevski et al., 2020).

Data Preparation for Large Language Models. Our EMG adaptor network is defined as

[1]LLaMA 2-7B
[2]LLaMA 3.2-3B

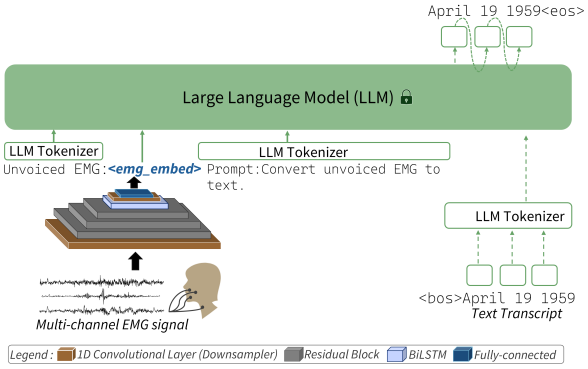


Figure 1: Our trainable EMG adaptor with frozen LLMs to transcribe text from only unvoiced EMG.

$\mathcal{G} : \mathbf{X^E} \rightarrow \mathbf{E}$. We contextualize the embeddings by prepending them with a text identifier $\mathbf{P_1}$ = `Unvoiced EMG:`, and appending a prompt describing the task as $\mathbf{P_2}$=`Prompt: Convert unvoiced EMG embeddings to text.` The LLM tokenizer converts the text identifier and prompt into text embeddings using the mapping $\mathcal{M} : \mathbf{X^P} \rightarrow \mathbf{H^P}$, where $\mathbf{X^P} = [\mathbf{P_1}, \mathbf{P_2}]$. To prepare the input for the LLM, the embeddings from the prompts are concatenated with the EMG embeddings.

Training Framework. For each unvoiced EMG signal, we have a corresponding text transcription $\mathbf{X^S}$. Following the recommendations of Gaddy and Klein (2021), we simplify our target by removing punctuation and converting all text to lowercase. We extract embeddings from the final LLM layer and compute the predicted logits $z_{t_s,y'}$ for each vocabulary item y' at position t_s. The cross-entropy loss over time steps is given by:

$$\mathcal{L} = -\sum_{t_s=1}^{T_s} \sum_{y' \in \mathcal{V}} y'_{t_s} \log \frac{\exp(z_{t_s,y'}/\tau)}{\sum_{v \in \mathcal{V}} \exp(z_{t_s,v}/\tau)},$$

where $\tau = 0.8$ is the temperature parameter and y'_{t_s} is the true class at t_s. We follow standard recommendations for fine-tuning LLMs, employing the AdamW optimizer (Loshchilov and Hutter, 2019) with a maximum learning rate of 5×10^{-5} and weight decay. During inference, we autoregressively generate (Dubey et al., 2024) the predicted target sequence with a beam-width (Freitag and Al-Onaizan, 2017) of 4. More implementation details are provided in Appendix B. Our codebase is release

3 Experimental Results and Discussion

Datasets. We primarily used the single-speaker, 8-channel, closed-vocabulary dataset from Gaddy

and Klein (2021), which comprises 67 words and approximately 26 minutes of unvoiced EMG data across 500 utterances. More details are provided in Appendix A. We accessed only the unvoiced EMG modality.

Baselines and Experimental Setup. We use the transducer model proposed by Gaddy and Klein (2021) as an application-specific baseline. Our analysis leverages two LLMs: Llama2-7B and Llama3-3B. We also fine-tune Llama3-3B using low-rank adaptation (LoRA) (Hu et al., 2022), training 0.13% of its parameters. We use three-fold validation and report word error rate (WER) statistics. Following standard practice, data is split 8:1:1 into training, validation, and test sets. All baselines are trained solely on unvoiced EMG. A minimal code implementation and sample predictions are included in the supplementary materials, with further details in Appendix B.

3.1 Key Findings

LLMs boost closed-vocabulary unvoiced EMG-to-text conversion by 30% with minimal data processing. As shown in Table 1, compared to the application-specific (App-Specific) model with transformers (54M trainable parameters), the proposed EMG adaptor (EMG-Ad) with frozen LLMs, using only 6M trainable parameters, achieves a 0.52 WER with raw EMG signals as input, outperforming the App-Specific model's 0.75 average WER. While the closed-vocabulary setting poses a challenge due to limited training data, it supports the hypothesis that LLMs, through large-scale training, have likely learned universal language representations that help understand unvoiced EMG with limited datasets. Fine-tuning offered only a 17% improvement over the App-Specific baseline in this setting, possibly due to overparameterization.

Handcrafted EMG features improve LLM performance for unvoiced EMG-to-text conversion in closed vocabulary. Leveraging recommendations from previous works on temporal (Jou et al., 2006) and spectral (Gaddy and Klein, 2020) features, we extract 112 time-varying EMG features and use them as inputs to the baselines. As shown in Table 1, these handcrafted features consistently outperform raw features across both LLMs used as inputs to the EMG adaptor, showing an average improvement of 15%. However, for the App-Specific baseline, raw features perform better, similar to Gaddy and Klein (2021)'s findings.

Table 1: Comparison of App-Specific models and EMG adaptors (EMG-Ad) with frozen and fine-tuned LLMs. Frozen parts are shown in *green*, and the best performance in each setting is in **bold**. Lower WER is better.

	Model	WER
Raw EMG	App-Specific(Gaddy and Klein, 2021)	0.75 ± 0.06
	EMG-Ad + *Llama2-7B*	0.65 ± 0.01
	EMG-Ad + *Llama3-3B*	**0.52 ± 0.05**
	EMG-Ad + Fine-tuned Llama3-3B	0.62 ± 0.04
Hand-crafted Features	App-Specific(Gaddy and Klein, 2021)	0.84 ± 0.06
	EMG-Ad + *Llama2-7B*	**0.49 ± 0.06**
	EMG-Ad + *Llama3-3B*	**0.49 ± 0.04**
	EMG-Ad + Fine-tuned Llama3-3B	0.55 ± 0.02

LLMs enable data-efficient learning for closed-vocabulary silent speech. To further evaluate the effectiveness of LLMs in facilitating learning from a limited number of samples, we randomly subsampled the training data from approximately 26 minutes to 6 minutes, as illustrated in Figure 2. Although the WER increases with the reduced training set, our LLM-based approach still outperforms the App-Specific baseline by an average of 26%. Evidenced by prior work (Diener et al., 2020a) and our pilot study in Section 3.3, surface-EMG signals exhibit distinct person-specific phenotypes. Thus, learning from a limited of samples facilitates building personalized silent-speech interfacing models for LLM assistants.

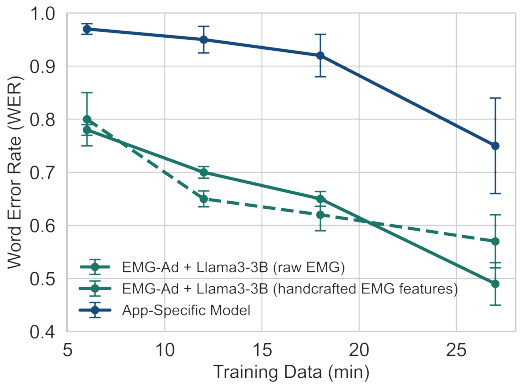


Figure 2: Performance of EMG adaptor with Llama3-3B model vs. App-Specific model across training dataset sizes for unvoiced raw EMG-to-text conversion.

Expanding LLMs to EMG is harder than audio. To demonstrate the potential of additional language modalities in expanding text-based LLMs, we adopt Ma et al. (2024)'s strategy of incorporating a speech encoder—both an end-to-end trained speech encoder using mel-frequency spectrogram features and a pretrained encoder (Baevski et al., 2020)—with a linear-projection head into

an LLM to convert speech to text (more details in Appendix B.3). LLMs learn 33% better from audio even with this simple approach, highlighting the overall task complexity of unvoiced EMG-to-text, as shown in Figure 3.

Augmenting training dataset with voiced EMG benefits specialized models more than LLMs. In an augmented setting, where we trained using voiced and unvoiced raw EMG, the additional modality led to a 20% improvement in the App-Specific model (expected behavior as Gaddy and Klein (2021)), while it offered little benefit to the LLM-based approach in this closed vocabulary setting as shown in Figure 3. This suggests that more dedicated efforts in instruction tuning or explicit pairing of voiced and unvoiced EMG (Xu et al., 2022) may be needed to learn improved representations from LLMs. However, in this paper, our focus remains on converting unvoiced EMG to text.

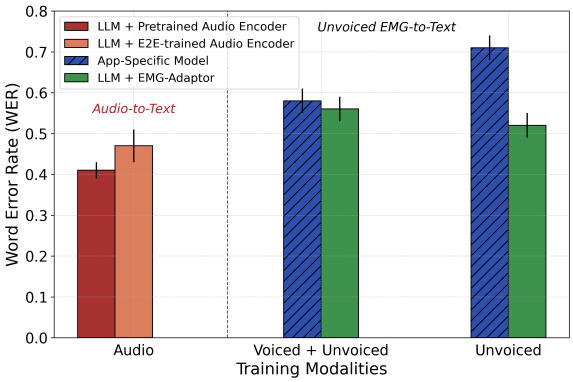


Figure 3: Performance comparison in (1) expanding LLMs to the audio vs. EMG modality, and in (2) training our LLM-based approach and the specialized model using voiced vs. unvoiced EMG data.

3.2 Ablation Analysis

Table 2 summarizes key variants of the EMG adaptor network design with raw unvoiced EMG, omitting downsampling via high-stride 1D CNN layers. Unlike specialized EMG-to-text models (Gaddy and Klein, 2021) that benefit from transformers, we find LSTMs perform better in this setting (Mohapatra et al., 2023c). This may be due to the shorter sequence length (average of four words per utterance with more details in Appendix A) in the closed-vocabulary dataset and our ability to leverage the language-pretrained transformer layers in LLMs. Appendix B.2 presents additional ablation results on the sequential backbone architectures.

Previous works with specialized models (Gaddy and Klein, 2021; Benster et al., 2024) generally optimize connectionist temporal classification (CTC) (Graves et al., 2006) loss with a high beam width (>100). However, most LLMs are decoder-only architectures and are trained with cross-entropy (CE) loss. One challenge in optimizing these embeddings using CTC loss is ensuring that their temporal length exceeds the target sequence length (Sudo et al., 2025) for stable optimization. To achieve this, we leverage 1D convolution layers to dilate the embeddings. However, we find that optimizing the embeddings extracted from these LLMs with CTC loss remains suboptimal compared to using CE loss with temperature and small beam widths of just 4 for inference.

Table 2: Ablation Study of EMG Adaptor training.

Component	Variants	WER
EMG-Adaptor w/ Llama3-3B	Fully Connected	0.70
	ResBlock (2)	0.64
	ResBlock (2) + Transformer	0.79
	ResBlock (2) + LSTM	0.53
Objective w/ Llama2-7B	Cross-Entropy (Section 2)	0.65
	CTC (Graves et al., 2006)	0.70

3.3 Further Explorations

Person-identification from unvoiced EMG with 96% accuracy. Physiological signals often carry person-specific phenotypes (Zlatintsi et al., 2023; Mohapatra et al., 2023b). To validate that EMG signals also encode such individualized traits, we conduct a pilot analysis on a public multi-subject dataset (Diener et al., 2020b), containing 1,000 unique utterances from four participants. We collapse the LLM's embeddings along the temporal dimension and train a simple classification head (details in Appendix C), achieving an average accuracy of 0.96. This strong performance, consistent with findings in other EMG settings (Diener et al., 2020a; Wand et al., 2009), reaffirms that unvoiced EMG exhibits distinct user-specific patterns—even when users speak the same text segment. To further support this, we also train a fully end-to-end (non-LLM) model for person identification, which achieves 0.99 accuracy, confirming that these signals are highly discriminative. Importantly, our goal is not to propose a state-of-the-art method for user identification using LLMs, but rather to highlight that unvoiced EMG signals inherently carry identifiable traits. This motivates the need for personalized modeling, which typically requires learning from limited data. In this context, our design choice—to use a lightweight trainable adaptor

with frozen LLMs—offers a practical and efficient solution. It enables learning user-specific representations with minimal data, a key requirement in personalized EMG-based systems. Our results further emphasize the importance of data-efficient learning, as demonstrated in Figure 2.

Exploration of data augmentation for unvoiced EMG. We explored two data augmentation schemes: (1) random temporal shifts per EMG channel (Gaddy and Klein, 2021, 2020) and (2) Hilbert-transform-based phase augmentation from limb-based EMG gesture recognition (Mohapatra et al., 2024c; Wang et al., 2025), but neither significantly improved performance. Developing tailored augmentation methods for unvoiced EMG—balancing diversity and phonetic coherence—could enhance learning from limited data.

Conclusion

Our approach demonstrates the potential of using LLMs to convert unvoiced EMG signals to text, achieving a 0.49 WER without any voiced data. In data-poor settings, it outperforms specialized models by 26%. Our experiments also highlight the value of hand-crafted features as input to LLMs for this task. This work[3] helps enable users who cannot produce vocal speech to interact with LLMs through unvoiced commands, especially as LLM-based assistants become ubiquitous.

Acknowledgments

We gratefully acknowledge the support from National Science Foundation grants 2038853, 2324936, and 2328032. We would like to thank the creators of the open-source dataset used in our study. Special thanks to Neil Zhang for his thoughtful feedback and discussions on language modeling methods.

Limitations

Exploration with open vocabulary. In most LLMs, the heavy lifting of supporting large vocabulary sizes (32,000 in Llama2 and 128,000 in Llama3) is carried out in the final layer (Wijmans et al., 2024). This makes learning even a closed vocabulary (67 words, approximately 4 words per utterance, with the potential for 50,000 unique utterances) from a new language modality a challenging task, which is the focus of this paper. Supporting a larger, in-the-wild open vocabulary using unvoiced EMG is even more complex, with its difficulty exacerbated by the overall lack of data and the need for personalized models. However, our current explorations lay the groundwork for exploring this direction next. Prior specialized approaches in this domain heavily rely on the availability of audio and voiced EMG, typically using CTC loss for optimization. This also presents an additional challenge in training LLMs in low-resource modalities with a new objective that does not effectively utilize their learned embeddings from large-scale data, leading to suboptimal training. These challenges can be addressed by exploring multimodal LLMs to leverage implicitly aligned embeddings or by reformulating the open vocabulary as a larger closed set and using target-steering methods to train on this restructured vocabulary (Han et al., 2024).

Broader Extension to more datasets and languages. Specialized EMG-to-text models are often designed for a predetermined EMG configuration or are deeply tied to phonetic auxiliary tasks, limiting them to the English language and a specific dataset. Our method has the potential to be extended to multilingual and multi-configuration instrumentation. However, due to the lack of publicly available closed-vocabulary datasets in such settings, we limit our investigation to English corpora. A systematic multilingual and diverse instrumentation closed- and open-set recording setup can accelerate exploration in this challenging direction of converting unvoiced EMG to text.

Additionally, our current approach is reliant on the embedding layers of LLMs, so it cannot work with language model APIs that do not provide direct access to these embeddings.

Ethical Concerns and Potential Risks

In this work, we utilize pretrained LLMs, specifically Llama, in accordance with their usage license, solely for academic research purposes. We do not foresee any immediate ethical concerns arising from our work. However, as an LLM application for interpreting biosignals, appropriate measures must be taken to preserve user privacy. Our techniques help make LLM-based assistants accessible to speech-impaired users, thereby encouraging socially beneficial outcomes.

[3] Our processed dataset, codebase and sample predictions are available at https://github.com/payalmohapatra/SilentSpeechLLM.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. Wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Tyler Benster, Guy Wilson, Reshef Elisha, Francis R Willett, and Shaul Druckmann. 2024. A cross-modal approach to silent speech with LLM-enhanced recognition. *arXiv preprint arXiv:2403.05583*.

Lorenz Diener, Shahin Amiriparian, Catarina Botelho, Kevin Scheck, Dennis Küster, Isabel Trancoso, Björn W. Schuller, and Tanja Schultz. 2020a. Towards silent paralinguistics: Deriving speaking mode and speaker id from electromyographic signals. In *Interspeech 2020*, pages 3117–3121.

Lorenz Diener, Mehrdad Roustay Vishkasougheh, and Tanja Schultz. 2020b. CSL-EMG_Array: An Open Access Corpus for EMG-to-Speech Conversion. In *Proceedings of Interspeech 2020*, pages 3745–3749.

Pranay Dighe, Yi Su, Shangshang Zheng, Yunshu Liu, Vineet Garg, Xiaochuan Niu, and Ahmed Tewfik. 2024. Leveraging large language models for exploiting asr uncertainty. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12231–12235. IEEE.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. *arXiv preprint arXiv:1702.01806*.

David Gaddy and Dan Klein. 2020. Digital voicing of silent speech. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5521–5530. Association for Computational Linguistics.

David Gaddy and Dan Klein. 2021. An improved model for voicing silent speech. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 175–181, Online. Association for Computational Linguistics.

Jose A Gonzalez-Lopez, Alejandro Gomez-Alanis, Juan M Martín Doñas, José L Pérez-Córdoba, and Angel M Gomez. 2020. Silent speech interfaces for speech restoration: A review. *IEEE access*, 8:177995–178021.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Chi Han, Jialiang Xu, Manling Li, Yi Fung, Chenkai Sun, Nan Jiang, Tarek Abdelzaher, and Heng Ji. 2024. Word embeddings are steers for language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16410–16430.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net.

Szu-Chen Jou, Tanja Schultz, Matthias Walliczek, Florian Kraft, and Alex Waibel. 2006. Towards continuous speech recognition using surface electromyography. In *Ninth International Conference on Spoken Language Processing*.

Wonjune Kang, Junteng Jia, Chunyang Wu, Wei Zhou, Egor Lakomkin, Yashesh Gaur, Leda Sari, Suyoun Kim, Ke Li, Jay Mahadeokar, et al. 2024. Frozen large language models can perceive paralinguistic aspects of speech. *arXiv preprint arXiv:2410.01162*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. 7th int. In *Conf. Learn. Represent. ICLR*.

Ziyang Ma, Guanrou Yang, Yifan Yang, Zhifu Gao, Jiaming Wang, Zhihao Du, Fan Yu, Qian Chen, Siqi Zheng, Shiliang Zhang, et al. 2024. An embarrassingly simple approach for LLM with strong ASR capacity. *arXiv preprint arXiv:2402.08846*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*.

Geoffrey S Meltzner, James T Heaton, Yunbin Deng, Gianluca De Luca, Serge H Roy, and Joshua C Kline. 2017. Silent speech recognition as an alternative communication device for persons with laryngectomy. *IEEE/ACM transactions on audio, speech, and language processing*, 25(12):2386–2398.

Geoffrey S Meltzner, James T Heaton, Yunbin Deng, Gianluca De Luca, Serge H Roy, and Joshua C Kline. 2018. Development of sEMG sensors and algorithms for silent speech recognition. *Journal of neural engineering*, 15(4):046031.

Payal Mohapatra, Ali Aroudi, Anurag Kumar, and Morteza Khaleghimeybodi. 2024a. Non-verbal hands-free control for smart glasses using teeth clicks. *arXiv preprint arXiv:2408.11346*.

Payal Mohapatra, Bashima Islam, Md Tamzeed Islam, Ruochen Jiao, and Qi Zhu. 2023a. Efficient stuttering event detection using siamese networks. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Payal Mohapatra, Shamika Likhite, Subrata Biswas, Bashima Islam, and Qi Zhu. 2024b. Missingness-resilient video-enhanced multimodal disfluency detection. In *Interspeech 2024*, pages 5093–5097.

Payal Mohapatra, Akash Pandey, Bashima Islam, and Qi Zhu. 2022. Speech disfluency detection with contextual representation and data distillation. In *Proceedings of the 1st ACM international workshop on intelligent acoustic systems and applications*, pages 19–24.

Payal Mohapatra, Akash Pandey, Sinan Keten, Wei Chen, and Qi Zhu. 2023b. Person identification with wearable sensing using missing feature encoding and multi-stage modality fusion. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–2. IEEE.

Payal Mohapatra, Akash Pandey, Yueyuan Sui, and Qi Zhu. 2023c. Effect of attention and self-supervised speech embeddings on non-semantic speech tasks. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9511–9515.

Payal Mohapatra, Lixu Wang, and Qi Zhu. 2024c. Phase-driven domain generalizable learning for nonstationary time series. *arXiv preprint arXiv:2402.05960*.

Vimal Mollyn, Riku Arakawa, Mayank Goel, Chris Harrison, and Karan Ahuja. 2023. Imuposer: Full-body pose estimation using imus in phones, watches, and earbuds. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

NICD. 2024. Quick statistics about voice, speech, language, and swallowing. Accessed February 13, 2025.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

R OpenAI. 2023. GPT-4 technical report. arxiv 2303.08774. *View in Article*, 2(5).

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Tanja Schultz and Michael Wand. 2010. Modeling coarticulation in EMG-based continuous speech recognition. *Speech Communication*, 52(4):341–353.

Tanja Schultz, Michael Wand, Thomas Hueber, Dean J Krusienski, Christian Herff, and Jonathan S Brumberg. 2017. Biosignal-based spoken communication: A survey. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12):2257–2271.

Tanmay Srivastava, Prerna Khanna, Shijia Pan, Phuc Nguyen, and Shubham Jain. 2024. Poster unvoiced: Designing an unvoiced user interface using earables and llms. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*, pages 871–872.

Yui Sudo, Muhammad Shakeel, Yosuke Fukumoto, Brian Yan, Jiatong Shi, Yifan Peng, and Shinji Watanabe. 2025. Joint beam search integrating ctc, attention, and transducer decoders. *IEEE Transactions on Audio, Speech and Language Processing*.

Weiting Tan, Hirofumi Inaguma, Ning Dong, Paden Tomasello, and Xutai Ma. 2024. SSR: Alignment-aware modality connector for speech language models. *arXiv preprint arXiv:2410.00168*.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2023. Salmonn: Towards generic hearing abilities for large language models. *arXiv preprint arXiv:2310.13289*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Michael Wand, Szu-Chen Stan Jou, Arthur R. Toth, and Tanja Schultz. 2009. Impact of different speaking modes on emg-based speech recognition. In *Interspeech 2009*, pages 648–651.

Lixu Wang, Bingqi Shang, Yi Li, Payal Mohapatra, Wei Dong, Xiao Wang, and Qi Zhu. 2025. Split adaptation for pre-trained vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Erik Wijmans, Brody Huval, Alexander Hertzberg, Vladlen Koltun, and Philipp Krähenbühl. 2024. Cut your losses in large-vocabulary language models. *arXiv preprint arXiv:2411.09009*.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430.

Zhiyang Xu, Ying Shen, and Lifu Huang. 2022. Multiinstruct: Improving multi-modal zero-shot learning via instruction tuning. *arXiv preprint arXiv:2212.10773*.

Jeonghun Yeo, Seunghee Han, Minsu Kim, and Yong Man Ro. 2024. Where visual speech meets language: VSP-LLM framework for efficient and context-aware visual speech processing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11391–11406, Miami, Florida, USA. Association for Computational Linguistics.

Wenyi Yu, Changli Tang, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2024. Connecting speech encoder and large language model for ASR. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12637–12641. IEEE.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128.

Ruidong Zhang, Mingyang Chen, Benjamin Steeper, Yaxuan Li, Zihan Yan, Yizhuo Chen, Songyun Tao, Tuochao Chen, Hyunchul Lim, and Cheng Zhang. 2021. Speechin: a smart necklace for silent speech recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(4):1–23.

Athanasia Zlatintsi, Panagiotis Paraskevas Filntisis, Niki Efthymiou, Christos Garoufis, George Retsinas, Thomas Sounapoglou, Ilias Maglogiannis, Panayiotis Tsanakas, Nikolaos Smyrnis, and Petros Maragos. 2023. E-prevention: The icassp-2023 challenge on person identification and relapse detection from continuous recordings of biosignals. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–2. IEEE.

APPENDIX

A Dataset Statistics

Table 3 provides a summary of the dataset statistics, while Figures 4 and 5 illustrate the distribution of the input EMG sequence lengths and target lengths.

Table 3: Comparison of dataset statistics from Gaddy and Klein (2020) and Diener et al. (2020b). Here, N represents the number of participants, and "Train" includes both training and validation samples.

Statistic	Gaddy	CSL
Dataset Size	500 utterances	1000 utterances
Target Type	Closed Vocabulary Template	Fixed Open Vocabulary
Number of Individuals (N)	1	4
Train Set (Train + Val)	450	800
Evaluation Set	50	200
k-fold Evaluations	3	3
Sampling Rate	1000 Hz → 800 Hz	2048 Hz

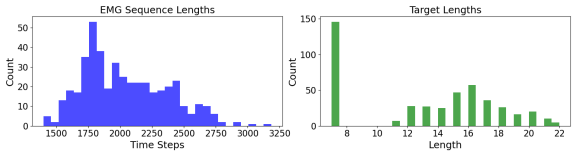


Figure 4: Histogram of the distribution of the sequence lengths and the target lengths for Gaddy and Klein (2020) closed-vocabulary dataset.

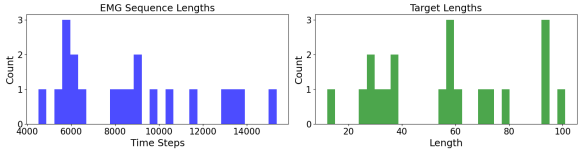


Figure 5: Histogram of the distribution of the sequence lengths and the target lengths for Diener et al. (2020b)'s Person 1 Block1 Initial segment of the data which is the superset for all the utterances.

B Implementation Details

We provide more implementation-specific details of the baselines here and summarize the sizes of all models in Table 4.

B.1 Application-specific Baseline

To test the application-specific baseline model under fair settings, we ran it with beam widths of 4 (same as our LLM-based model) and 100. As noted in Table 5, the WER for both the beam widths are nearly the same, indicating that beam width has a negligible effect on the prediction in the case of

Table 4: Trainable parameters of different models. Values are reported in millions (M) and thousands (k).

Model	Trainable Parameters
Application-Specific Model	54M
Application-Specific (Modified)	8.1M
LLaMA-7B + EMG Adaptor, LLaMA-3B + EMG Adaptor	6.4M
Fine-tuned LLaMA-3B + EMG Adaptor	10.3M
Fine-tuned LLaMA-3B + EMG Adaptor (Alternative)	5.94M
LLaMA-3B + Audio Adaptor	590k

the closed vocabulary. We also train the baseline model with smaller trainable parameters (8M) by reducing the feature size, size of feedforward linear layers, and number of layers in the transformers to 512, 512, and 2 respectively. As shown in Table 5, the WER with a smaller baseline model is also in the same range as with 54M parameters. This highlights the fact that the difference in performance between our model and the baseline model in Table 1 is due to the choice of LLM for the prediction.

Table 5: WER performance comparison of the original (54M) and reduced (8M) Gaddy models under different beam-width settings.

Model	Beam-width	WER
Original (54M)	$n = 4$	0.70
	$n = 100$	0.72
Reduced (8M)	$n = 4$	0.87

B.2 Additional Ablation: EMG-Adaptor Backbone Variants

We conducted additional ablation studies to explore different sequential backbones for the EMG-Adaptor (EMG-Ad) when used with the LLaMA 3B LLM. The results are summarized in Table 6.

Table 6: Performance of different sequential backbones for the EMG-Adaptor with LLaMA 3B. Number of transformer layers are denoted as L.

Sequential Backbone	Params	WER
BiLSTM	5.94M	0.52
LSTM	5.5M	0.58
Transformer (6L + Sinusoidal) (Vaswani et al., 2017)	6.7M	0.79
Transformer (6L + RoPE) (Touvron et al., 2023)	6.7M	0.75
Transformer (2L + RoPE)	5.3M	0.72

We experimented with Rotary Position Encoding (RoPE) (Touvron et al., 2023), motivated by its effectiveness in the LLaMA models, with the intuition that it might produce better input tokens for the LLM. Although RoPE-based transformers improved performance over the vanilla sinusoidal variant, they still underperformed compared to LSTM-based models.

This aligns with findings from prior time-series literature. For example, Zeng et al. (2023) have argued that the permutation-invariant nature of self-attention leads to temporal information loss. Similarly, practical studies such as IMUPoser (Mollyn et al., 2023) empirically corroborate that LSTMs outperform transformers in fine-grained time-series tasks like human activity recognition, e.g., stating in Section 4.1 of (Mollyn et al., 2023): *"Although we did experiment with newer architectures such as transformers, we found these models did not perform well in practice."*.

While advanced positional encoding schemes and specialized architectures (Wu et al., 2021; Nie et al., 2022) have enhanced transformer performance on time series, we restrict our analysis to vanilla transformers for simplicity and focus on their ability to generate suitable EMG tokens for LLMs. Our findings suggest that LSTMs are better suited for this task. This observation is currently limited to short sequences drawn from a closed vocabulary. Future work will investigate more specialized transformer-based architectures for unvoiced EMG modeling.

B.3 Expanding LLMs to Audio

In this experiment, we primarily leverage the implementation of Tang et al. (2023) and the idea proposed by Ma et al. (2024) to employ an embarrassingly simple approach for speech recognition with LLMs, using a linear projector from a frozen speech encoder. We use the wav2vec 2.0 (Baevski et al., 2020; Mohapatra et al., 2023a, 2024b, 2022) BASE architecture as our speech encoder, which produces a 768-dimensional feature vector. This vector is then passed through two linear layers to generate the 3072-dimensional input required for Llama3-3B. While the speech encoder can be replaced with alternatives such as HuBERT (Hsu et al., 2021) or Whisper (Radford et al., 2023), our goal is not to optimize speech-to-text conversion. Instead, we aim to demonstrate that while both audio and EMG involve expanding an LLM's capability to a new language modality, integrating EMG signals poses significantly greater challenges.

B.4 Additional Reproducibility Information

All experiments were conducted using NVIDIA A100 GPUs (3 available, 40GB CUDA memory) and TITAN RTX GPUs (4 available, 24GB CUDA memory) with a maximum runtime of 12 hours in PyTorch. Hyperparameter tuning combined manual and random search, typically requiring fewer than five runs, with selection based solely on validation loss and WER.

The batch size for LLM experiments was 8, and the maximum number of epochs was 500. For the App-Specific model, we retain the original settings (Gaddy and Klein, 2021), where the authors re-batchified the input by rolling the temporal dimensions to support training on longer sequences.

C Person Identification using Unvoiced EMG : Implementation Details

For the person identification task, we use LLaMA 3.2-3B. Overall the model architecture remains the same as shown in Figure 1 except we do not use any prompt in this case. The logits output from the LLM, $Z \in \mathbb{R}^{T \times \mathcal{V}}$, is reduced to $\overline{Z} \in \mathbb{R}^{1 \times \mathcal{V}}$ by taking mean along the time axis. The $\overline{Z}$ is then fed into several linear layers to predict logits $y^p \in \mathbb{R}^{1 \times 4}$ as there are 4 distinct persons in the dataset (Diener et al., 2020b). Due to the linear layers, the number of trainable parameters is 38M, which is higher than the numbers mentioned in Table 4 for LLM-based models. Using y^p and actual person ID, the softmax loss is calculated to fine-tune 32M parameters.

D Additional Ethical Statements

In preparing this work, we only used AI assistants in the capacity to polish the language in the manuscript.

Decoder-Only LLMs can be Masked Auto-Encoders

Dan Qiao[1,2], **Yuan Gao**[2] **Zheming Yang**[2], **Di Yang**[2],
Ziheng Wu[2], **Pengcheng Lu**[2], **Minghui Qiu**[2], **Juntao Li**[1*] **Min Zhang**[1]
[1]School of Computer Science and Technology, Soochow University
[2]ByteDance Inc
qiaodan.23@bytedance.com

Abstract

Modern NLP workflows (e.g., RAG systems) require different models for generation and embedding tasks, where bidirectional pre-trained encoders and decoder-only Large Language Models (LLMs) dominate respective tasks. Structural differences between models result in extra development costs and limit knowledge sharing between tasks. In this work, we present UniMAE, a novel unsupervised training method that transforms a Decoder-Only LLM into a **Uni**-Directional **M**asked **A**uto-**E**ncoder. UniMAE compresses high-quality semantic information into the [EOS] embedding while preserving the generation capabilities of LLMs. Comprehensive evaluations across 56 MTEB datasets demonstrate that UniMAE can achieve state-of-the-art results under unsupervised settings with merely 100 training steps, establishing the first effective approach to unifying generation and representation learning in decoder-only architectures.

1 Introduction

Pre-trained language models have been widely applied in various scenarios (Devlin, 2018; Ouyang et al., 2022; Brown et al., 2020; Wu et al., 2025; **?**). However, the workflows of real-world applications like RAG (Fan et al., 2024) or retrieval systems (Zhu et al., 2023) often involve collaboration between generation tasks and embedding tasks. For example, RAG requires retrieval followed by generation (Asai et al., 2023), while retrieval systems often need to rewrite queries before performing the search (Ma et al., 2023; Liu and Mozafari, 2024).

For embedding tasks, encoder models with bidirectional attention mechanisms have been the mainstream choice (Devlin, 2018; Liu, 2019). More recently, decoder-only large language models (LLMs) with unidirectional attention have demonstrated the ability to compress knowledge from trillions of tokens during pre-training (Radford, 2018; Ouyang et al., 2022; Guo et al., 2025), thus have been widely utilized in various generative tasks. This structural separation leads to extra training and deployment costs and impedes knowledge sharing between tasks (Asai et al., 2023). In the long run, enabling large language models (LLMs) to perform both generation and embedding tasks simultaneously holds significant promise.

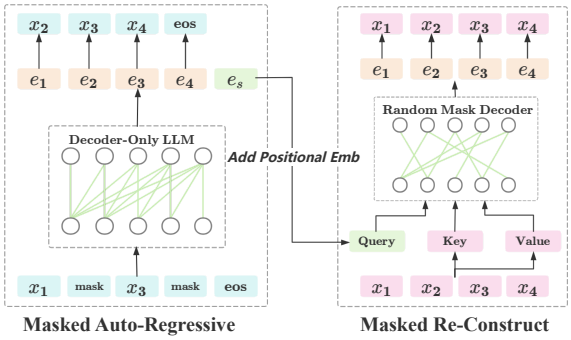


Figure 1: Overview of UniMAE training. The left part shows the Masked Auto-Regressive process, while the right part shows the reconstruction process using a tiny decoder with [EOS] embeddings to recover the input.

In this paper, we introduce UniMAE, an unsupervised method that compresses the semantic content of input text into the [EOS] (end-of-sequence) embedding of the last transformer layer, while preserving the capabilities of original LLMs (Figure 1). We scale our method across models with 1B, 3B, and 8B parameters and conduct evaluations on 56 datasets of the Massive Text Embeddings Benchmark (MTEB) (Muennighoff et al., 2022). UniMAE achieves state-of-the-art performance on MTEB under an unsupervised setting within only 100 training steps. Evaluations on language modeling tasks demonstrate that UniMAE not only enhances representation capabilities but also maintains the generation abilities of LLMs. Domain post-training experiments indicate that UniMAE can simultaneously improve the generation and embedding performance in a single training process.

* Corresponding author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 713–723
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

To our knowledge, UniMAE is the first effective approach to unifying generation and representation learning in decoder-only architectures.

2 Method

2.1 Motivation

Unlike pre-trained encoder models that utilize special tokens like [CLS] for sentence embeddings, LLMs primarily focus on predicting the next token during pre-training and lack a dedicated position for sentence representation. As a result, neither averaging the token embeddings nor directly using the [EOS] embedding can effectively represent the entire sentence. (Springer et al., 2024; Jiang et al., 2023; BehnamGhader et al., 2024).

To bridge this gap, recent studies have focused on enhancing the representational capabilities of large models. Many directly use the model as an encoder, employing average embeddings or token-level embeddings for supervised contrastive fine-tuning (Lee et al., 2024; Li et al., 2024). Springer et al. (2024) and BehnamGhader et al. (2024) argue that unidirectional attention prevents each token from incorporating information from subsequent tokens, thereby limiting the model's contextual understanding. Springer et al. (2024) addresses this limitation by repeating the sentence twice. Meanwhile, BehnamGhader et al. (2024) applies a Bidirectional Attention Mechanism and adapts the MLM task to LLM, then finally uses average pooling to get sentence embeddings.

However, these methods still fail to address the lack of a unified pooling mechanism in LLMs. Moreover, they simply repurpose LLMs as encoders during fine-tuning, thereby overlooking the potential loss of their generative capabilities.

2.2 Overall framework

Similar to Visual-MAE (He et al., 2022), UniMAE employs the [EOS] embedding as the latent representation of the corrupted input and initializes a small decoder to reconstruct the original input based on this representation. Unlike other approaches, our method maintains a unidirectional attention mechanism characteristic of large language models (LLMs). We illustrate the entire training process in Figure 1.

The overall objective consists of two components: 1. Masked Auto-Regressive (MAR) and 2. Masked Re-Construct (MRC). The complete training process involves jointly optimizing two sets of parameters: the LLM parameters Φ_{llm} and small decoder parameters Φ_{dec}.

2.3 Masked Auto-Regressive (MAR)

The sole distinction between the classic Auto-Regressive (Radford, 2018; Radford et al., 2019; Brown et al., 2020) and MAR is that the input is corrupted with random masks of ratio p_{mar}. For a sentence X, it is corrupted by replacing some tokens with mask tokens, resulting in the masked sentence $\tilde{X}$. The MAR loss can be expressed as:

$$L_{MAR} = \sum_t CE(x_t \mid \tilde{x}_{i<t}; \Phi_{llm}), \quad (1)$$

where $x_i \in X$ is the i-th token or the original text, $\tilde{x}_i \in \tilde{X}$ is the i-th token of the corrupted text. Φ_{llm} denotes the parameters of the original LLM.

2.4 Masked Re-Construct (MRC)

We believe that the representation at [EOS] position has the potential to encapsulate the semantic meaning of the entire input. Since only the last token can attend to all the input tokens under the unidirectional attention mechanism. So the latent representation h_{eos} is:

$$h_{eos} \leftarrow \Phi_{llm}\left(\tilde{X}\right) \quad (2)$$

The Masked Re-Construct (MRC) process utilizes a newly initialized small decoder, which employs the [EOS] embedding combined with partial input information to reconstruct the original sentence. The input of the tiny decoder can be formulated as:

$$\begin{aligned} H_1 &\leftarrow [h_{eos} + p_0, \ldots, h_{eos} + p_N], \\ H_2 &\leftarrow [h_{eos}, e_{x_1} + p_1, \ldots, e_{x_N} + p_N], \end{aligned} \quad (3)$$

where p_i is the trainable positional embedding, e_i is the input token embedding. The only difference between MRC and traditional cross-attention is that we applied a mask to the attention matrix, forcing the model to rely more on the h_{eos} for sentence recovery. We set a mask ratio p_{mrc} and let $B_{ij} \sim \text{Bernoulli}(p_{mrc}) \in \{0, 1\}$. Then the computation process of attention can be expressed with the following formula:

$$\begin{aligned} Q = H_1 W^Q, K = H_2 W^K, V = H_2 W^V \\ M_{ij} = \begin{cases} 0, & \text{if } i \neq j \wedge B_{ij} = 1 \\ -\infty, & \text{else} \end{cases}, \\ A = \text{softmax}\left(\frac{Q^T K}{\sqrt{d}} + M\right) V. \end{aligned} \quad (4)$$

Attn.	Categories →	Retr.	Rerank.	Clust.	PairClass.	Class.	STS	Summ.	Avg
	# of datasets →	15	4	11	3	12	10	1	56
				LLaMA3.2-1B					
Uni.	MEAN	15.13	42.96	34.30	53.50	56.20	56.27	28.06	39.32
Uni.	ECHO	24.57	48.27	36.68	67.65	65.60	71.86	29.79	48.28
Bi.	MNTP	18.53	43.54	34.20	56.30	60.24	61.15	27.21	42.12
Uni.	UniMAE	**34.48**	50.51	38.42	75.91	68.88	73.14	29.93	52.81
Bi.	MNTP + SimCSE	30.59	49.67	37.48	74.41	65.70	72.85	**30.71**	50.73
Uni.	UniMAE + SimCSE	31.54	**54.23**	**43.22**	**78.04**	**69.26**	**76.95**	30.14	**54.11**
				LLaMA3.2-3B					
Uni.	MEAN	14.61	43.22	34.73	55.50	55.55	56.00	22.13	39.09
Uni.	ECHO	25.44	48.64	36.86	70.68	67.96	71.95	24.31	49.16
Bi.	MNTP	17.75	44.69	36.06	57.47	59.79	61.87	23.06	42.38
Uni.	UniMAE	34.25	51.79	38.95	78.31	70.14	73.06	29.25	53.30
Bi.	MNTP + SimCSE	34.35	50.91	39.83	76.21	66.00	73.77	**30.64**	52.61
Uni.	UniMAE + SimCSE	**36.09**	**55.41**	**43.77**	**80.90**	**71.60**	**76.53**	30.33	**56.11**
				LLaMA3.1-8B					
Uni.	MEAN	14.26	46.89	37.84	70.12	66.73	62.32	24.98	42.95
Uni.	ECHO	25.53	49.75	37.77	70.11	67.03	71.83	27.32	49.24
Bi.	MNTP	22.83	48.46	40.96	64.26	60.11	64.50	27.40	45.95
Uni.	UniMAE	35.37	52.74	41.25	78.23	68.49	73.45	30.40	53.87
Bi.	MNTP + SimCSE	**39.75**	53.59	42.70	78.71	67.38	**76.84**	**31.77**	55.81
Uni.	UniMAE + SimCSE	38.51	**56.91**	**46.06**	**80.15**	**68.77**	76.21	29.80	**56.60**

Table 1: Results on 56 MTEB datasets with best results highlighted in bold, and the second-best results underlined.

The MRC object can be formalized as:

$$\mathcal{L}_{MRC} = \sum_{t} CE(x_t \mid A, H_1, H_2, \Phi_{dec}). \quad (5)$$

The joint optimization of MAR and MRC is expressed as follows:

$$\mathcal{L}_{UniMAE} = \alpha \mathcal{L}_{MAR} + \mathcal{L}_{MRC}, \quad (6)$$

where α is the weight to control the MAR objective.

3 Experiment

3.1 Settings

For embedding tasks, we conduct evaluations on the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2022), a collection of 7 diverse embedding task categories covering a total of 56 datasets. All evaluations are done using the official MTEB code repository. For preliminary experiments and ablations, we use MTEB-15 a subset including 15 representative tasks. Details are attached in Appendix A.

For baselines, we compare our method with average pooling, Echo (Springer et al., 2024), which repeats the sentence twice and averages the token embedding of the second sentence, and LLM2Vec (BehnamGhader et al., 2024), which sequentially employs MNTP and SimCSE (Gao et al., 2021b) to train the model. To make a comparison with LLM2Vec with SimCSE training, we also provide UniMAE with further SimCSE training. We provide training details in Appendix C.

3.2 Embedding Performance

Table 1 shows that, after UniMAE + SimCSE training, we observe a considerable improvement in performance for all three models compared with direct mean pooling, which is 28% for the 1B model, 43% for the 3B model and 32% for the 8B model. Using only UniMAE can significantly surpass all single unsupervised learning methods, UniMAE even outperforms MNTP + SimCSE training on 1B and 3B models. Overall, the UniMAE framework achieves the state-of-the-art results across models of all scales. Scores for each dataset are shown in Appendix G.

3.3 Will UniMAE hurt LLMs?

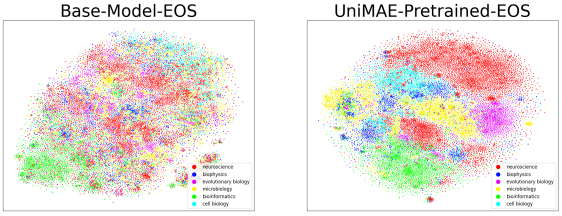
We selected 10 commonly used datasets for evaluating generative language models, including ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019) and MMLU (Hendrycks et al., 2021). Details can be found in Appendix E. As shown in Figure 3b, after UniMAE training, the generative capabilities of the model are nearly equivalent to those of the original foundation models. We also tested the baseline method, MNTP training, which modifies causal attention to bidirectional attention. We observe that the structural modification significantly

reduces the model's performance on language modeling tasks. The same situation occurs in supervised / unsupervised contrastive learning applied to the representations after average pooling.

3.4 UniMAE optimizes Vector Space

We apply t-SNE transformations to the top 6 classes' sentence embeddings from BiorxivClusteringS2S dataset, using PCA initialization, 2000 iterations, and early exaggeration of 20.

Figure 2 shows the distribution of the UniMAE and baseline embeddings. UniMAE improves spatial distribution by bringing samples closer within each group using [EOS] token embeddings. This improved distribution highlights the effectiveness of our approach in boosting LLM performance as an encoder.



(a) Base Model (b) UniMAE

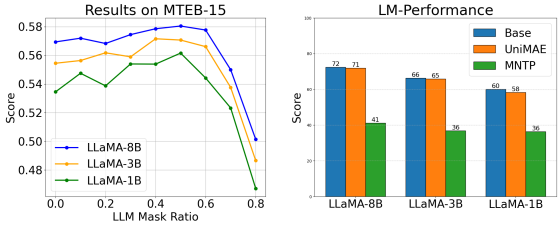
Figure 2: t-SNE visualization of sentence embeddings from top 6 classes on BiorxivClusteringS2S testset.

3.5 Is MAR necessary?

We designed MAR to prevent the model from learning the shortcut of simply memorizing all the input tokens during training. Therefore, we introduce noise to the input, allowing the model to extract semantic representations even from incomplete inputs, rather than simply memorizing them. To prove that MAR is superior to AR, we trained different models using input masking ranging from 0% to 80% and performed inference on MTEB-15. As shown in Figure 3a, for models of different sizes, the optimal mask ratio falls between 40% and 60%. For convenience, we set this ratio to 50% for all models. In the detailed results, we observe that input masking significantly improved performance on tasks related to semantic similarity, such as Clustering and STS, while the enhancement for retrieval tasks was relatively minor.

3.6 Domain Post-Training Result

We believe that UniMAE can not only enhance the representation ability of LLMs but also serve as a novel pre-training and post-training method to



(a) Mask Ratio Search (b) LM Evaluation

Figure 3: (a) Results under different MAR mask ratio on MTEB-15. (b) Performance on language model tasks.

Categories → **# of sub-tasks →**	**LM.** 3	**Retr.** 7	**Class.** 7
Qwen2-0.5B	71.12	36.91	81.43
Qwen2-0.5B + AR	73.16	35.89	83.98
Qwen2-0.5B + UniMAE	**74.55**	**50.84**	**95.38**

Table 2: Model performance after domain post-training

simultaneously improve both generative and representational capabilities. We conduct E-Commerce domain post-training for 100B tokens, we describe more details in the Appendix F. The evaluation contains three types of tasks: domain language modeling, retrieval, and classification. Language model tasks (LM.) are primarily QA, retrieval (Retr.), and classification (Class.) tasks that are similar to those in MTEB. Table 2 demonstrates that UniMAE training slightly outperforms Auto-Regressive in generation tasks, and significantly outperforms Auto-Regressive in embedding tasks.

4 Conclusion

We introduce UniMAE, a method capable of training models for both generation and embedding tasks. We evaluate UniMAE on diverse datasets, including open-source and domain-specific corpora, across both types of tasks. UniMAE achieves state-of-the-art performance on the MTEB benchmark without compromising the language modeling capabilities of LLMs. As a domain post-training approach, UniMAE enhances both the representational and generative capacities of models simultaneously. In future work, we plan to apply UniMAE for large-scale pre-training from scratch and instruction alignment, aiming to develop a new generation of LLMs that excel in both generation and embedding tasks.

5 Limitations

There is an inherent limitation: we initialized a small decoder from scratch to reconstruct the original input. However, during the inference process, we directly discarded this small decoder and instead utilized the results generated by the LLM for both generation and representation tasks. This approach leads to a certain degree of parameter waste. If this method is scaled up for large-scale zero pretraining, such waste would be suboptimal. In the future, we will explore new structural designs to address this issue.

Acknowledgments

We want to thank all the anonymous reviewers for their valuable comments. This work was supported by the National Science Foundation of China (NSFC No. 62206194), the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20220488), the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001), and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021a. A framework for few-shot language model evaluation.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. 2017. Dbpedia-entity v2: A test collection for entity search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 1265–1268. ACM.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2023. Scaling sentence embeddings with large language models. *arXiv preprint arXiv:2307.16645*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Chaofan Li, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Yingxia Shao, Defu Lian, and Zheng Liu. 2024. Making text embedders few-shot learners. *arXiv preprint arXiv:2409.15700*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3622–3628.

Jie Liu and Barzan Mozafari. 2024. Query rewriting via large language models. *arXiv preprint arXiv:2403.09060*.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *arXiv preprint arXiv:2305.14283*.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2024. Repetition improves language model embeddings. *arXiv preprint arXiv:2402.15449*.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106.

Ziheng Wu, Zhenghao Chen, Ruipu Luo, Can Zhang, Yuan Gao, Zhentao He, Xian Wang, Haoran Lin, and Minghui Qiu. 2025. Valley2: Exploring multimodal models with scalable vision-language design. *arXiv preprint arXiv:2501.05901*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.

A Massive Text Embeddings Benchmark (MTEB)

The original MTEB benchmark (Muennighoff et al., 2022) consists of 56 datasets of various embedding tasks. Some tasks like classification and clustering require additional use of small clustering models or classification heads after extracting embeddings. For a fair comparison, we use the official MTEB code repository [1] to test all the models, simply overriding the encode function to extract representations for the given text. Since the models under evaluation have different architectures and pooling methods, they may have different optimal instructions for their respective tasks. For fair comparisons, we do not use any instruction and only extract embeddings from the test texts.

MTEB consists of diverse small and large embedding tasks. Some datasets like MSMARCO (Nguyen et al., 2016) and DBPedia (Hasibi et al., 2017) have even more than 5,000,000 samples. Considering the test speed, for preliminary experiments and ablation studies, we select 15 representative tasks as a subset in Table 3, which aligns with LLM2VEC (BehnamGhader et al., 2024). For each

[1] https://github.com/embeddings-benchmark/mteb

task category, we selected the same proportion of datasets to ensure that the results in MTEB-15 are not biased towards MTEB-56.

Category	Dataset
Retrieval (3)	SciFact ArguAna NFCorpus
Reranking (2)	StackOverflowDupQuestions SciDocsRR
Clustering (3)	BiorxivClusteringS2S MedrxivClusteringS2S TwentyNewsgroupsClustering
Pair Classification (1)	SprintDuplicateQuestions
Classification (3)	Banking77Classification EmotionClassification MassiveIntentClassification
STS (3)	STS17 SICK-R STSBenchmark
Overall	15 datasets

Table 3: Datasets of MTEB-15

B Usage Instructions

In the entire framework, there are two important hyper-parameters: the mask ratio p_{mar} of MAR, and the weight α of MAR. Based on our experience, a higher p_{mar} often provides greater benefits for tasks such as STS and clustering, which are sensitive to vector space comparisons, while the improvement for retrieval tasks is relatively smaller. For the weight $alpha$, it depends on your expectations for the LLM base. If you aim for the LLM itself to learn from the data (wiki-text), you can set a relatively high α. However, if your goal is to enhance the representation capability while maintaining the original abilities of the LLM, a value of 0.1 would be sufficient.

Regarding the number of training steps, we only trained for 100 steps on general open-source data. This is mainly because LLMs have typically encountered this open-source data during their pre-training. Numerous studies have shown that performing autoregressive training on repetitive data can easily lead to model collapse. Therefore, if there is data that can enhance the LLM itself, it can be trained extensively, and full-parameter training can be employed. For instance, we conducted full-parameter training on domain-specific data with 100 billion tokens, which continuously improved the model's overall capabilities. However, if training on data does not benefit LLMs or even hurt LLMs, we recommend using LoRA to converge new parameters instead.

C Unsupervised Training Settings

C.1 Baselines

For Echo (Springer et al., 2024), we simply repeat the input text and take the average output embedding of the second sentence as the global embedding. For MNTP and MNTP + SimCSE (BehnamGhader et al., 2024), we train the LLaMA-3.2-1B and LLaMA-3.2-3B using the official code with the setting mentioned for Sheared-LLaMA-1.3B, as they do not release models based on these two foundation models. For LLaMA-3.1-8B, we directly use the released lora weights [2].

C.2 Our method

We keep the same training settings across all the models. For UniMAE training, we train all the models using Peft [3] package with LORA (Hu et al., 2021) adapter to train the models, with $\alpha_{lora} = 32, r = 16, lr = 1e-4$. We only add trainable lora weight to the attention block and mlp block. We train the model on randomly selected wiki-text data [4] for 100 steps with a global batch size of 32 samples. The max length of each sample is 512. The learning rate is $1e-4$ and we use a constant learning rate scheduler. The MAE weight α is set to 0.1 and MRC weight β is set to 1. MAR mask ratio p_{mar} is set to 0.5, and MRC mask ratio p_{mrc} is set to 0.5.

For SimCSE training, we train all the models using Peft package with LORA (Hu et al., 2021) adapter to train the models, with $\alpha = 32, r = 16, lr = 1e-4$. We only add trainable lora weight to the attention block and mlp block. We train the model on randomly selected wiki-data for 100 steps with a global batch size of 64 samples. The max length of each sample is 512. The learning rate is $1e-4$ and we use a constant learning rate scheduler.

D Combine UniMAE with Supervised Contrastive Learning

Experiments in table 1 have shown that UniMAE significantly improves the embedding performance

[2] https://huggingface.co/McGill-NLP/LLM2Vec-Meta-Llama-3-8B-Instruct-mntp

[3] https://github.com/huggingface/peft

[4] https://huggingface.co/datasets/wikimedia/wikipedia

Categories →	Retr.	Rerank.	Clust.	PairClass.	Class.	STS	Avg
# of datasets →	3	2	3	1	3	3	15
			LLaMA3.2-1B				
Base Model	51.17	66.08	37.23	93.76	69.70	81.85	63.05
UniMAE + SimCSE	**53.95**	**67.17**	**38.45**	**94.57**	**70.41**	**84.00**	**64.62**
			LLaMA3.2-3B				
Base Model	50.77	66.18	**40.69**	88.89	67.53	79.81	62.51
UniMAE + SimCSE	**56.24**	**69.32**	40.23	**94.25**	**70.28**	**83.50**	**65.58**
			LLaMA3.1-8B				
Base Model	53.07	67.19	40.21	95.04	69.42	82.44	64.32
UniMAE + SimCSE	**57.93**	**70.67**	**42.65**	**96.13**	**70.77**	**84.51**	**67.00**

Table 4: Results on MTEB-15 after supervised contrastive learning.

of LLMs. Many studies have proven that fine-tining supervised contrastive learning data can further improve the performance of LLMs on embedding tasks. To see whether our method can further improve the performance under supervised settings, we collect open-sourced supervised contrastive learning data to train the base model and model trained using UniMAE + SimCSE. We follow the setting of BGE-ICL (Li et al., 2024), which also does not modify the model architecture, using [EOS] embedding as the sentence embedding for contrastive learning. We use the official training code of BGE-ICL [5]. Data can be found here [6]. For base models and models trained using UniMAE + SimCSE, we both use the [EOS] embedding of the last transformer layer as the sentence embedding to conduct supervised contrastive learning. We train each model for 4000 steps with a global batch size of 64 samples. The max length of each text is set to 512. We use the Peft package with LORA (Hu et al., 2021) adapter to train the models. The learning rate is $1e-4$ and we use a constant learning rate scheduler.

The result can be found in table 4. It demonstrates that our approach not only directly enhances the base model's capabilities but also shows improved performance after additional contrastive learning compared to the original base model.

E Language Model Evaluation

We use the lm-evaluation-harness package (Gao et al., 2021a) to evaluate the LLMs on 10 downstream tasks: ARC-E (Clark et al., 2018), LAMBADA (Paperno et al., 2016), LogiQA (Liu et al., 2020), PIQA (Bisk et al., 2020), SciQ (Welbl et al., 2017), and WinoGrande (Sakaguchi et al., 2021), HellaSwag (Zellers et al., 2019), ARC-C (Clark et al., 2018)), Natural Questions (Kwiatkowski et al., 2019), MMLU (Hendrycks et al., 2021). We keep a zero-shot setting for each task.

F E-commerce Domain Post-training

In the E-commerce domain, the pre-training dataset consists of approximately 100 billion tokens (Due to the principle of anonymity, we cannot disclose the specific platform). To maintain a certain level of general capability for the LLM, we incorporated a substantial amount of open-source data, with domain-specific data accounting for 25% of the total. This domain-specific data mainly includes ASR and OCR from product live streams and short videos, user comments, and textual information about the products themselves. We set $p_{mar} = 0$, $p_{mrc} = 0.5$, $\alpha = 1$, where MAR has already degraded into AR. This is because: 1. we need to enhance the model's overall domain knowledge. 2. The downstream representation tasks only involve retrieval and classification, while MAR offers greater improvements for STS and clustering tasks.

The testing scenarios primarily focus on governance within the E-commerce Platform, with the ultimate goal of determining whether a given sample violates the platform's regulatory guidelines. The tasks can be categorized into three main types. The language model task primarily involves generative QA, where for each sample, the model is directly asked whether it violates a specific set of regulations. The retrieval task focuses on identifying samples that share the same type of violation as a given problematic sample. The classification task involves using a large language model (LLM) to offline extract text representations of a sample, which serve as input for a classification head to categorize the sample's violations.

[5] https://github.com/FlagOpen/FlagEmbedding
[6] https://huggingface.co/datasets/cfli/bge-full-data

G Full Result

Table 5 and table 6 shows the detailed scores of models trained with UniMAE and UniMAE + SimCSE on all of the 56 MTEB datasets.

Task	LLaMA3.2-1B	LLaMA3.2-3B	LLaMA3.1-8B
AmazonCounterfactualClassification.	76.58	80.66	75.01
AmazonPolarityClassification.	78.49	74.89	76.79
AmazonReviewsClassification.	40.26	39.98	40.34
ArguAna.	48.28	50.38	53.87
ArxivClusteringP2P.	45.54	46.99	47.45
ArxivClusteringS2S.	34.58	36.62	38.63
AskUbuntuDupQuestions.	55.73	57.02	55.00
BIOSSES.	79.97	84.28	85.12
Banking77Classification.	74.76	79.21	76.55
BiorxivClusteringP2P.	36.41	35.89	35.86
BiorxivClusteringS2S.	26.53	26.61	29.71
CQADupstackTexRetrieval.	18.02	18.95	17.59
ClimateFEVER.	21.99	19.03	22.36
DBPedia.	26.13	22.56	23.78
EmotionClassification.	44.82	48.18	42.86
FEVER.	40.32	35.14	37.44
FiQA2018.	24.50	26.62	25.70
HotpotQA.	43.58	49.07	50.59
ImdbClassification.	70.94	74.79	74.78
MSMARCO.	21.33	20.29	19.64
MTOPDomainClassification.	92.25	93.61	93.58
MTOPIntentClassification.	73.41	74.20	71.03
MassiveIntentClassification.	70.48	70.32	68.05
MassiveScenarioClassification.	74.79	75.03	74.79
MedrxivClusteringP2P.	29.50	26.77	29.28
MedrxivClusteringS2S.	22.85	23.61	25.98
MindSmallReranking.	29.92	29.14	32.91
NFCorpus.	26.40	28.92	28.06
NQ.	33.42	32.80	33.70
QuoraRetrieval.	80.71	81.28	82.67
RedditClustering.	43.36	43.28	48.13
RedditClusteringP2P.	58.50	59.13	60.81
SCIDOCS.	12.70	12.43	13.51
SICK-R.	71.41	67.37	68.52
STS12.	59.38	60.37	58.89
STS13.	75.44	75.00	73.64
STS14.	68.58	69.69	70.45
STS15.	78.73	79.10	78.93
STS16.	76.94	78.43	75.52
STS17.	83.87	80.87	83.13
STS22.	63.52	63.75	64.68
STSBenchmark.	73.52	71.77	75.59
SciDocsRR.	72.72	74.21	78.48
SciFact.	63.71	65.22	66.17
SprintDuplicateQuestions.	81.65	90.50	87.58
StackExchangeClustering.	57.92	63.54	63.80
StackExchangeClusteringP2P.	32.58	33.96	33.36
StackOverflowDupQuestions.	43.68	45.80	44.58
SummEval.	29.93	29.25	30.40
TRECCOVID.	44.99	40.94	44.71
Touche2020.	11.16	10.19	10.74
ToxicConversationsClassification.	69.44	72.38	69.21
TweetSentimentExtractionClassification.	60.38	58.43	58.90
TwentyNewsgroupsClustering.	34.79	32.05	40.71
TwitterSemEval2015.	61.70	60.61	62.45
TwitterURLCorpus.	84.37	83.82	84.65
AVG	52.81	53.30	53.87

Table 5: Unsupervised results of UniMAE transformed models on MTEB.

Task	LLaMA3.2-1B	LLaMA3.2-3B	LLaMA3.1-8B
AmazonCounterfactualClassification	72.39	78.01	67.97
AmazonPolarityClassification	76.94	79.38	77.04
AmazonReviewsClassification	37.80	40.27	40.27
ArguAna	41.26	49.76	52.12
ArxivClusteringP2P	47.43	48.03	49.04
ArxivClusteringS2S	39.29	40.66	45.22
AskUbuntuDupQuestions	57.32	59.21	59.41
BIOSSES	83.21	85.32	87.23
Banking77Classification	78.85	81.68	79.57
BiorxivClusteringP2P	37.60	35.43	36.64
BiorxivClusteringS2S	33.39	33.62	34.74
CQADupstackTexRetrieval	16.13	21.40	19.48
ClimateFEVER	15.81	19.54	22.67
DBPedia	24.52	24.85	29.53
EmotionClassification	48.30	50.47	45.34
FEVER	24.78	37.28	55.28
FiQA2018	23.43	27.64	28.81
HotpotQA	33.02	46.53	53.72
ImdbClassification	74.79	77.36	76.43
MSMARCO	16.98	19.99	21.56
MTOPDomainClassification	93.54	94.85	93.28
MTOPIntentClassification	70.46	75.96	71.61
MassiveIntentClassification	71.56	71.67	70.35
MassiveScenarioClassification	76.89	78.67	77.22
MedrxivClusteringP2P	30.06	29.83	30.58
MedrxivClusteringS2S	29.34	28.56	30.46
MindSmallReranking	32.83	31.90	33.36
NFCorpus	27.36	31.99	27.99
NQ	27.32	30.93	31.83
QuoraRetrieval	85.61	85.71	85.97
RedditClustering	54.29	57.34	57.39
RedditClusteringP2P	57.13	59.43	61.87
SCIDOCS	14.53	15.74	16.89
SICK-R	73.99	70.22	72.73
STS12	65.70	63.51	62.53
STS13	81.08	79.88	79.11
STS14	75.14	74.36	74.75
STS15	83.15	82.50	81.50
STS16	81.18	81.66	80.75
STS17	88.88	86.53	85.55
STS22	55.87	61.67	58.93
STSBenchmark	81.32	79.68	78.99
SciDocsRR	80.06	80.85	84.22
SciFact	61.10	65.41	67.24
SprintDuplicateQuestions	85.35	91.35	88.71
StackExchangeClustering	66.02	68.15	71.84
StackExchangeClusteringP2P	32.44	33.70	33.75
StackOverflowDupQuestions	46.71	49.69	50.67
SummEval	30.14	30.33	29.80
TRECCOVID	51.78	52.27	51.29
Touche2020	9.45	12.26	13.27
ToxicConversationsClassification	69.14	71.11	67.81
TweetSentimentExtractionClassification	60.45	59.74	58.39
TwentyNewsgroupsClustering	48.39	46.73	55.11
TwitterSemEval2015	65.35	65.59	67.52
TwitterURLCorpus	83.42	85.75	84.24
Average	54.11	56.11	56.60

Table 6: Unsupervised results of UniMAE+SimCSE transformed models on MTEB.

Mitigating Posterior Salience Attenuation in Long-Context LLMs with Positional Contrastive Decoding

Zikai Xiao[1,5*] Ziyang Wang[2*] Wen Ma[1] Yan Zhang[3]
Wei Shen[1] Yan Wang[1] Luqi Gong[4†] Zuozhu Liu[1,5†]
[1]Zhejiang University, [2]University of Science and Technology of China, [3]Bytedance,
[4]Zhejiang Lab, [5]Zhejiang Key Laboratory of Medical Imaging Artificial Intelligence
zikai@zju.edu.cn

Abstract

While Large Language Models (LLMs) support long contexts, they struggle with performance degradation within the context window. Current solutions incur high training costs, leaving statistical behaviors and cost-effective approaches underexplored. From the decoding perspective, we identify the Posterior Salience Attenuation (PSA) phenomenon, where the salience ratio correlates with long-text performance degradation. Notably, despite the attenuation, gold tokens still occupy high-ranking positions in the decoding space. Motivated by it, we propose the training-free Positional Contrastive Decoding (PCD) that contrasts the logits derived from long-aware attention with those from designed local-aware attention, enabling the model to focus on the gains introduced by large-scale short-to-long training. Through the analysis of long-term decay simulation, we demonstrate that PCD mitigates attention score degradation. Experimental results show that PCD achieves state-of-the-art performance on long-context benchmarks.

1 Introduction

The maximum context lengths of large language models (LLMs) have steadily increased, yet their effective utilization remains limited: most open-source models experience sharp performance degradation beyond 16k tokens (Hsieh et al., 2024a; Hengle et al., 2024; Zhang et al., 2024a). Previous works have sought to explain and improve context utilization. From empirical observations derived from the generated text, the "lost in the middle" effect Liu et al. (2024); Zhang et al. (2024a) reveals inconsistent performance drops across different positions (Zhang et al., 2024a), and the "Know but Don't Tell" phenomenon reveals that while models encode target information, they fail to leverage it in generating accurate responses (Lu et al., 2024), a gap also examined in theoretical models of reasoning (Bi et al., 2025).

To fully utilize the context, data-driven methods have been proposed, such as synthetic key-value retrieval mechanisms (An et al., 2024) and multi-document question-answering frameworks (DATA). In terms of model design, Tworkowski et al. (2024) enhances attention layers with external memory, while Zhang et al. (2024b) leverages Multi-scale Positional Encoding for capturing multi-scale distance awareness. These approaches often involve costly annotation and training. For inference-time methods, Segment Reranking (Dsouza et al., 2024; Peysakhovich and Lerer, 2023) addresses the "lost in the middle" problem at the prompt level by rearranging key segments to the beginning and end of the context. Additionally, (Hsieh et al., 2024b) estimates and calibrates positional bias using an auxiliary priori dummy document. Prompt-dependent solutions remain brittle and highly sensitive to the hyper-specific prompt formulation. Thus, devising quantitative analysis and cost-effective solutions for long context utilization remains challenging.

In this work, we uncover the Posterior Salience Attenuation (PSA) phenomenon and corresponding Positional Contrastive Decoding (PCD) to make LLMs effectively utilize the context. Through analysis in decoding-space (Section 2.1), we found that the posterior salience of the gold label gradually degrades as context length grows, when controlling tasks to maintain consistent difficulty. Despite the posterior salience decreases, the ranking of the gold label often remains among the top ranks (top 0.006% in Fig. 3 (a), regarding the entire vocabulary (Dubey et al., 2024)), suggesting a decoding strategy that amplifies the salience of gold token. Through the analysis of numerous error cases, we found that the model tends to adopt tokens that are closer to the query (proximal tokens), leading to incorrect responses. To enhance the distance

[*]Co-first author.
[†]Co-corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 724–733
July 27 - August 1, 2025

awareness of LLMs, we proposed PCD that contrasts the logits derived from standard attention scores with those from designed local-aware attention, enabling the model to focus on the gains introduced by large-scale short-to-long training. We progressively induce excessive rotation beyond the default angles from high to low frequencies to construct the local-aware attention, inspired by insights from frequency analysis in RoPE (Su et al., 2024). The numerical and theoretical analysis and experimental analysis demonstrate that PCD effectively enhances long-awareness.

2 Methods

In long-context scenarios, denote $f(\theta)$ as LLM with parameters θ. Given a sequence of tokens $\mathbf{x} = [x_1, ..., x_L]$ from the input context, the model $f(\theta)$ generates the next token y^* by computing the posterior probability distribution $P_{f(\theta)}(y^*|\mathbf{x}_{\leq L})$, where $\mathbf{x}_{\leq L}$ represents the input sequence.

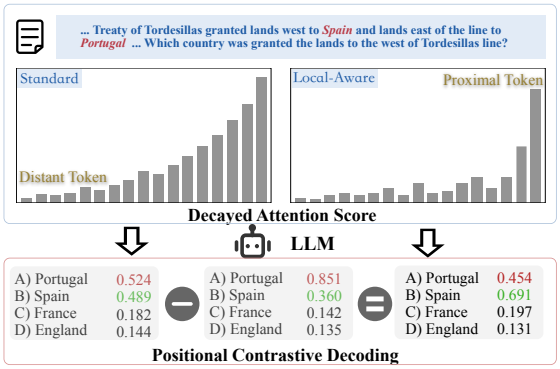


Figure 1: An illustration of PCD, contrasting logits from long-aware and local-aware attention, and amplifying the gains of large-scale short-to-long training.

2.1 Posterior Salience Attenuation in Long Context LLMs

PSA describes a statistical rank decrease of the gold token y^* in the model's decoding space. as the context length L increases. Concretely, for an input $x_{\leq L}$, Let Q denote the query set, $\mathcal{V}$ the vocabulary, and y_i^* the gold token for the i-th query. we define the *salience score*:

$$S(L) = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{1 + \sum_{v \in \mathcal{V}} \mathbb{I}\left(P_{f(\theta)}(v|x_{\leq L}^{(i)}) > P_{f(\theta)}(y_i^*|x_{\leq L}^{(i)})\right)} \quad (1)$$

where $\mathbb{I}(\cdot)$ is the indicator function, which evaluates to 1 if the condition inside is true and 0 otherwise. The $S(L)$ quantifies the extent to which the model prioritizes y_i^* by counting the number of tokens $v \in \mathcal{V}$ whose predicted probabilities exceed that of y_i^*. It then computes the reciprocal of this count to reflect the rank of y_i^* and averages these reciprocal ranks across all queries in Q. A decreasing trend in $S(L)$ with increasing L indicates a weakening of the model's ability to prioritize the gold token as the context length grows.

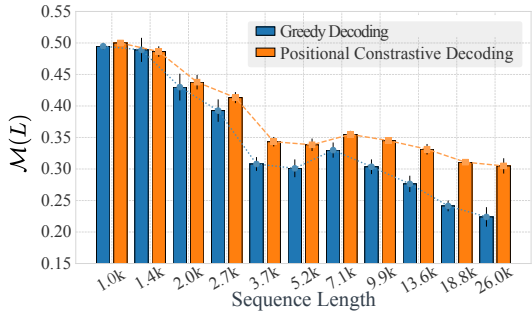


Figure 2: PCD effectively alleviates the decrease in salience scores with increasing input length.

Fig. 2 illustrates the salience score per incorrect sample as context length increases in the key-value retrieval task. As the context length grows, the salience score of incorrect predictions decreases. Notably, the gold label consistently ranks highly, typically within the top 8 tokens in decoding space, even for vocabularies as large as 128k, as shown in Fig. 3 (a). This suggests that while the model positioned the gold label within a high rank, it failed to select it due to insufficient confidence, underscoring the need for decoding strategies that boost its prominence.

2.2 Positional Contrastive Decoding

PCD contrasts logits from standard (long-aware) attention with those from a designed local-aware attention. This aims to mitigate PSA and counteract proximal bias by amplifying long-distance signals relative to local ones, thereby better leveraging the model's short-to-long context training. Let d denote the embedding dimension, and $j \in \{1, 2, ..., d/2\}$ index the rotation blocks in the position encoding matrix. The total number of rotation blocks is $d/2$, where each block processes a 2D subspace of the embedding space.

Standard Logits Based on Rotary Position Embedding (RoPE) (Su et al., 2024), which encodes positional information through rotational transformations, we compute the position-aware query and key vectors for a token at position m with embedding $\mathbf{x}_m \in \mathbb{R}^d$ as:

$$\mathbf{q}_m = R_{\Theta,m}^d \mathbf{W}_q \mathbf{x}_m, \quad \mathbf{k}_n = R_{\Theta,n}^d \mathbf{W}_k \mathbf{x}_n \quad (2)$$

where $R_{\Theta,m}^d \in \mathbb{R}^{d \times d}$ is a block-diagonal rotation

matrix composed of $d/2$ orthogonal sub-blocks:

$$R^d_{\Theta,m} = \bigoplus_{j=1}^{d/2} \begin{pmatrix} \cos m\theta_j & -\sin m\theta_j \\ \sin m\theta_j & \cos m\theta_j \end{pmatrix} \quad (3)$$

The angular frequency θ_j follows a geometric progression $\theta_j = B^{-2(j-1)/d}$, establishing the long-term decay property of attention scores from proximal to distant tokens. The standard logits are then obtained as $\mathbf{L} = f_\theta(\mathbf{q}_m, \mathbf{k}_n)$.

Local-Aware Logits Inspired by the decay properties of positional encodings—where high-frequency components focus on local differences and low-frequency components capture global patterns—we induce over-rotation into RoPE's low-frequency encodings, which encourages the model to be more sensitive to local details. We lower the base frequency from B to B' to update the frequencies $\theta'_j = (B')^{-2(j-1)/d}$, then gradually increase the rotation from high-frequency to low-frequency position encodings via a transition function $T(x) = 2 - \exp(\alpha x), \quad x \in [0, 1]$, where $x = \frac{j}{d/2}$ maps the block index $j \in \{0, 1, \ldots, d/2\}$ to the normalized interval $[0, 1]$. The modified angular frequencies are computed as:

$$\theta^*_j = T\left(\frac{j}{d/2}\right)\theta_j + \left(1 - T\left(\frac{j}{d/2}\right)\right)\theta'_j \quad (4)$$

Using the over-rotated matrix R^*_m that incorporates θ^*_j, the perturbed logits are computed as:

$$\mathbf{L}^* = f\left(R^*_m \mathbf{W}_q \mathbf{x}_m, R^*_n \mathbf{W}_k \mathbf{x}_n\right) \quad (5)$$

Contrastive Logits We combine the standard logits $\mathbf{L}$ and over-rotated logits $\mathbf{L}^*$ using a contrastive mechanism that operates on the top γ tokens (ranked by probability) in $\mathbf{L}_0$. The contrastive logits are computed as:

$$\tilde{\mathbf{L}} = (1 + \beta)\mathbf{L} - \beta\mathbf{L}^* \quad (6)$$

where $\beta > 0$ controls the contrast intensity, and $\gamma \in \mathbb{N}$ determines the number of top tokens to which the contrastive mechanism is applied.

2.3 Spectral Analysis of Contrastive Decoding with RoPE

The contrastive decoding operation enhances long-range attention through spectral interference. By introducing over-rotated low-frequency components, the modified attention spectrum $S_{\text{cd}}(k)$ slows the decay rate of attention scores by a factor of $(\ln B / \ln B')^{2/d}$, where B and B' are the bases of the original and perturbed angular frequencies, respectively. Furthermore, the contrastive coefficient β amplifies the original decay curve by incorporating the influence of long-term, slow-decaying logits. For a detailed derivation, see Appendix A.1.

3 Experiments and Analysis

3.1 Experiment Setup

Models and Datasets. We utilize Llama3-8B-8k model (Dubey et al., 2024), the long-context fine-tuned variants Llama3-8B-262k and Llama3-8B-1048k by Gradient AI. We tested tasks from **RULER**(Hsieh et al., 2024a), **InfiniteBench**(Zhang et al., 2024a). We test 262k variant on **LongBench**(Bai et al., 2024).

Baselines. For comparison, we employed decoding methods including greedy search, beam search, and DoLa (Chuang et al., 2024), along with training-free calibration techniques such as MsPoE (Zhang et al., 2024b), Segment-Reranking (Dsouza et al., 2024; Peysakhovich and Lerer, 2023), and Rephrasing (Zhang et al., 2024a; Yu, 2023).

3.2 Comparison Results

As shown in Table 2, PCD consistently enhances model performance across varying context lengths in retrieval tasks without additional training. Beam search also improves performance, while rephrasing does not alter the model's inherent retrieval ability. Segment Reranking (SegR) is task-dependent, offering benefits only when semantic order is less critical. In real-world tasks (Table 3), both MsPoE and PCD demonstrate stable performance. However, rephrasing is sensitive to evaluation methods, and SegR struggles in complex semantic contexts due to its reliance on simpler ordering patterns.

Table 1: Ablation study of PCD hyperparameters on Infinite Bench (accuracy %, variance %). Adjusted each hyperparameter individually from the optimal set and computed mean and variance over 3 runs.

Parameter	Tested Range	Recommended	Optimal	Acc. (%)	Variance (%)
Base (w/o PCD)	–	–	–	72.00	
Transition (α)	[0.1, 0.5]	0.1–0.2	0.2	78.50	1.2
Contrast (β)	[1.0, 4.0]	1.5–2.5	2.5	77.90	3.1
Frequency ($\frac{B'}{B}$)	[1e-6, 1e-1]	1e-4	1e-4	75.80	2.3
Top-γ	[10, 200]	20–30	30	71.50	1.8

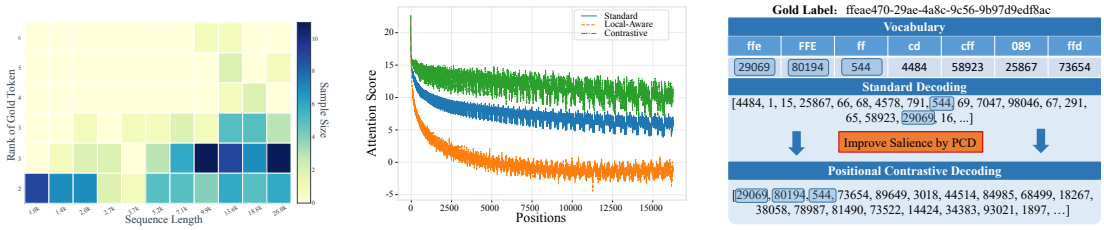
Table 2: Performance Comparison on Tasks from RULER and InfiniteBench

Model (Max Context)	Method	InfiniteBench: KV Retrieval (Accuracy %)			RULER: Variable Tracking (F1 Score)		
		4k	8k	16k	4k	8k	16k
Llama-3-8B (8k)	Base	97.6	92.4	–	66.38	57.93	–
	PCD	**100** (↑2.4%)	**91.0** (↓1.4%)	–	**68.91** (↑2.53)	**60.07** (↑2.14)	–
Llama-3-8B (262k)	Base	89.2	72.0	52.0	74.02	71.21	64.40
	Beam-Search	89.0	77.0	53.0	74.34	71.25	65.08
	DoLa-Low	92.0	76.0	53.0	76.48	72.93	66.48
	DoLa-High	93.0	76.0	54.0	77.19	72.87	67.29
	MsPoE	90.0	72.0	51.0	74.29	70.03	65.10
	SegR	93.0	76.0	54.0	0.0	0.0	0.0
	Rephrasing	92.0	73.0	50.0	81.60	79.28	70.56
	PCD	**92.0** (↑2.8%)	**79.0** (↑7.0%)	**55.0** (↑3.0%)	**81.80** (↑7.78)	**77.92** (↑6.71)	**69.04** (↑4.64)
Llama-3-8B (1048k)	Base	94.0	92.0	84.0	67.15	72.78	65.21
	PCD	**95.0** (↑1.0%)	**96.0** (↑4.0%)	**87.0** (↑3.0%)	66.77 (↓0.38)	71.19 (↓1.59)	**69.11** (↑3.9)

Notes: 1) Model variants are distinguished by their maximum context lengths (e.g., 8k, 262k, 1048k). 2) Arrows indicate PCD's relative change from the baseline. 3) Metrics: Key-Value Retrieval (Accuracy%); Variable Tracking (F1 Score).

Table 3: Performance Comparison on LongBench

Method	Multifieldqa_zh	Narrativeqa	Multifieldqa_en	2wikimqa	Qasper	Musique	Hotpotqa	Avg
Base	46.72	20.03	51.27	15.50	26.26	6.87	15.22	25.98
MsPoE	50.02	18.96	51.39	13.97	24.86	7.59	17.16	26.27
SegR	4.86	4.18	27.41	10.13	26.41	3.94	8.31	12.18
Rephrasing	45.13	18.94	49.53	13.22	28.70	6.25	13.28	25.02
PCD	**51.09** (↑4.37)	**20.31** (↑0.28)	50.11 (↓1.16)	**16.47** (↑0.97)	**27.13** (↑0.87)	6.70 (↓0.17)	**15.29** (↑0.07)	**26.87** (↑0.89)



(a) (b) (c)

Figure 3: (a) Distribution of gold label ranks across samples; (b) Single Layer Analysis: PCD mitigates salience attenuation by decelerating long-term decay; PCD mitigates long-term attention decay by slowing the salience ratio degradation; (c) Case Study: PCD enhances the ranks of answer-related tokens.

3.3 Hyperparameter Ablation

The hyperparameter ablation on InfiniteBench is evaluated in Table 1. Optimal performance is achieved with $\beta = 2.5$, which scales the preference for long-range awareness. Moderate frequency perturbation ($\frac{B}{B'} = 10^4$) performs best. The transition function coefficient α and top-γ are stable and generally require no adjustment.

3.4 Simulating Long-Term Decay of PCD

We employed single-layer attention to investigate how PCD mitigates long-range attention decay. We substrated logits of standard RoPE (with $B = 10^6$) against those generated from an over-rotated variant (with $B' = 10^4$). We applied a smooth transition function ($\alpha = 0.4$) and a contrastive coefficient $\beta = 0.6$. Attention scores were computed for a sequence length of $16,384$ with an embedding dimension $d = 512$. The results in Fig. 3 (b) demonstrate that the over-rotated variant exhibits sharper decay for local modeling, while PCD mitigates long-range decay, enhancing global awareness. To further validate the effectiveness of PCD in slowing down attention score decay, we conducted additional simulations under varying head dimensions, frequency parameters B', and contrastive coefficients β. For more details, refer to Section A.1.

3.5 Qualitative Study

We demonstrate the phenomenon of PSA, how PCD recalibrates the logits when the model fails to predict the gold label, as shown in Fig. 3 (c). Greedy decoding may prioritize irrelevant tokens, but with PCD, logits are recalibrated to boost the rank of the correct token and its related variants. More details can be found in A.3.

3.6 Conclusion

We investigate the performance degradation of long-context LLMs and identify the Posterior Salience Attenuation (PSA) phenomenon. To mitigate this, we propose Positional Contrastive Decoding (PCD), a cost-effective method that amplifies the salience of the gold token via contrastive logits. Experimental results show PCD achieves consistent performance on long-context benchmarks.

Limitations

PCD cannot extend the model's attention window. It also shows minor improvement for short-text tasks and may vary in effectiveness depending on the positional encoding design. Additionally, hybrid contrastive decoding between different series of models and embedding models in Retrieval-Augmented Generation (RAG) systems remains underexplored.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 62476241), the Natural Science Foundation of Zhejiang Province, China (Grant No. LZ23F020008), Zhejiang Key Laboratory of Medical Imaging Artificial Intelligence, and Zhejiang University–Angelalign Inc. R&D Center for Intelligent Healthcare.

References

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, and Jian-Guang Lou. 2024. Make your llm fully utilize the context. *arXiv preprint arXiv:2404.16811*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Jinhe Bi, Danqi Yan, Yifan Wang, Wenke Huang, Haokun Chen, Guancheng Wan, Mang Ye, Xun Xiao, Hinrich Schuetze, Volker Tresp, et al. 2025. CoT-Kinetics: A theoretical modeling assessing lrm reasoning process. *arXiv preprint arXiv:2505.13408*.

Ta-Chung Chi, Ting-Han Fan, Peter J Ramadge, and Alexander Rudnicky. 2022. Kerple: Kernelized relative positional embedding for length extrapolation. *Advances in Neural Information Processing Systems*, 35:8386–8399.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

TUNING ON SYNTHETIC DATA. From artificial needles to real haystacks: Im-proving retrieval capabilities in llms by fine-tuning on synthetic data.

Amanda Dsouza, Christopher Glaze, Changho Shin, and Frederic Sala. 2024. Evaluating language model context windows: A" working memory" test and inference-time correction. *arXiv preprint arXiv:2407.03651*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Amey Hengle, Prasoon Bajpai, Soham Dan, and Tanmoy Chakraborty. 2024. Multilingual needle in a haystack: Investigating long-context behavior of multilingual large language models. *arXiv preprint arXiv:2408.10151*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024a. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long T Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, et al. 2024b. Found in the middle: Calibrating positional attention bias improves long context utilization. *arXiv preprint arXiv:2406.16008*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Taiming Lu, Muhan Gao, Kuai Yu, Adam Byerly, and Daniel Khashabi. 2024. Insights into llm long-context failures: When transformers know but don't tell. *arXiv preprint arXiv:2406.14673*.

Alexander Peysakhovich and Adam Lerer. 2023. Attention sorting combats recency bias in long context language models. *arXiv preprint arXiv:2310.01427*.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. 2024. Focused transformer: Contrastive training for context scaling. *Advances in Neural Information Processing Systems*, 36.

Yijiong Yu. 2023. " paraphrasing the original text" makes high accuracy long-context qa. *arXiv preprint arXiv:2312.11193*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, et al. 2024a. Infinite bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277.

Zhenyu Zhang, Runjin Chen, Shiwei Liu, Zhewei Yao, Olatunji Ruwase, Beidi Chen, Xiaoxia Wu, and Zhangyang Wang. 2024b. Found in the middle: How language models use long contexts better via plug-and-play positional encoding. *arXiv preprint arXiv:2403.04797*.

A Appendix

A.1 Theoretical Analysis of Contrastive Decoding with RoPE

Definition A.1 (RoPE Transformation)**.** *For embedding dimension $d = 2h$, position indices $m, n \in \mathbb{N}$, and word embeddings $x_m, x_n \in \mathbb{R}^d$, the RoPE-formulated query and key vectors are:*

$$\begin{aligned} q_m &= R_\Theta(m) W_q x_m, \\ k_n &= R_\Theta(n) W_k x_n \end{aligned} \tag{7}$$

where the block-diagonal rotation matrix $R_\Theta(m) \in \mathbb{R}^{d\times d}$ consists of h orthogonal sub-blocks:

$$\begin{aligned} R_\Theta(m) &= \bigoplus_{j=1}^{h} R_{\theta_j}(m), \\ R_{\theta_j}(m) &= \begin{pmatrix} \cos m\theta_j & -\sin m\theta_j \\ \sin m\theta_j & \cos m\theta_j \end{pmatrix} \end{aligned} \tag{8}$$

with angular parameters $\theta_j = B^{-2(j-1)/d}$.

Lemma A.2 (Spectral Representation)**.** *The attention score between positions m and n admits:*

$$S(k) \triangleq q_m^\top k_n = \sum_{j=1}^{h} A_j \cos(k\theta_j + \phi_j) \tag{9}$$

where $k = m - n$, $A_j = \|(W_q x_m)_j\| \|(W_k x_n)_j\|$, and ϕ_j is the phase difference between the j-th components of $W_q x_m$ and $W_k x_n$ in polar coordinates.

Proof. For each 2D subspace $V_j = \mathbb{R}^2$, let $(q_j^{(1)}, q_j^{(2)}) = R_{\theta_j}(m)(W_q x_m)_j$ and similarly for k_j. Using the inner product invariance under rotation:

$$\begin{aligned} S_j(k) &= \langle R_{\theta_j}(m)(W_q x_m)_j,\ R_{\theta_j}(n)(W_k x_n)_j \rangle \\ &= \langle (W_q x_m)_j,\ R_{\theta_j}(k)^\top (W_k x_n)_j \rangle \\ &= A_j \cos(k\theta_j + \phi_j) \end{aligned} \tag{10}$$

Summing over all subspaces gives the complete spectrum.

Definition A.3 (Perturbed RoPE)**.** *Define a frequency-perturbed variant with modified base $B' = 10^2$, yielding angular parameters:*

$$\theta'_j = (B')^{-2(j-1)/d}, \quad j = 1, ..., h \tag{11}$$

with corresponding attention scores:

$$S'(k) = \sum_{j=1}^{h} A_j \cos(k\theta'_j + \phi_j) \tag{12}$$

Lemma A.4 (Decay Characterization)**.** *For $k > B^{2/d}$, the original scores satisfy:*

$$|S(k)| \le C_1 k^{-d/2} e^{-k^{2/d} \ln B} \tag{13}$$

where $C_1 = \sum_{j=1}^{h} A_j (B^{-2(j-1)/d})^{d/2}$.

Proof. Split the sum at critical index $j_0 = \lceil \frac{d}{2} \ln(k)/\ln B \rceil$:

$$|S(k)| \le \underbrace{\sum_{j=1}^{j_0} A_j \cos(k\theta_j)}_{\text{(I)}} + \underbrace{\sum_{j=j_0+1}^{h} A_j \cos(k\theta_j)}_{\text{(II)}} \tag{14}$$

Term (I) decays algebraically:

$$\sum_{j=1}^{j_0} A_j \le C_1 k^{-d/2} \tag{15}$$

Term (II) decays exponentially using $|\cos x| \le e^{-x^2/2}$ for $x \ge 1$:

$$\sum_{j=j_0+1}^{h} A_j e^{-(k\theta_j)^2/2} \le C_2 e^{-k^{2/d} \ln B} \tag{16}$$

Combining both terms gives the dominant algebraic decay.

Theorem A.5 (Contrastive Decoding Enhancement)**.** *The contrastive scores* $S_{CD}(k) = (1+\lambda)S(k) - \lambda S'(k)$ *satisfy:*

$$\limsup_{k\to\infty} \frac{\ln|S_{CD}(k)|}{\ln|S(k)|} \leq \left(\frac{\ln B'}{\ln B}\right)^{2/d} < 1 \quad (17)$$

Proof. Expanding the contrastive scores:

$$S_{\text{CD}}(k) = \sum_{j=1}^{h} A_j \left[(1+\lambda)\cos(k\theta_j) \quad -\lambda\cos(k\theta'_j)\right] \quad (18)$$

For low-frequency components ($j \leq j_0$), expand using perturbation $\delta_j = \theta'_j - \theta_j$:

$$\begin{aligned}\Delta_j(k) &\triangleq \cos(k\theta_j) - \cos(k\theta'_j) \\ &\approx k\delta_j \sin(k\theta_j) + \frac{(k\delta_j)^2}{2}\cos(k\theta_j)\end{aligned} \quad (19)$$

The leading term preserves oscillatory behavior while introducing linear growth:

$$|S_{\text{CD}}(k)| \geq \lambda \sum_{j=1}^{j_0} A_j k\delta_j |\sin(k\theta_j)| - O(k^2) \quad (20)$$

Combining with Lemma A.4 establishes the improved decay rate.

A.2 Long-term Decay Simulation under Hyperparameter

To comprehensively evaluate the impact of PCD on long-term attention decay, we conducted a series of numerical simulations under various hyperparameter settings. These simulations aimed to analyze how different configurations of embedding dimension d, contrastive coefficient β, transition function coefficient α, and base frequencies B and B' influence the decay rate of attention scores.

In our experiments, we systematically varied the embedding dimension d to examine its impact on the decay dynamics. As shown in Fig. 4, larger values of d generally lead to slower decay rates, as the increased dimensionality allows for more nuanced representations of positional information. However, the benefits of larger d diminish beyond a certain threshold, highlighting the trade-off between model capacity and computational efficiency.

The contrastive coefficient β plays a crucial role in balancing the influence of standard and perturbed logits. Our simulations reveal that moderate values of β (e.g., $\beta = 2.5$) yield the most effective decay mitigation, as they sufficiently amplify long-range awareness without introducing excessive noise. Extreme values of β, on the other hand, either fail to enhance long-term attention or disrupt the model's local modeling capabilities.

The transition function coefficient α governs the smoothness of the transition between original and perturbed frequencies. Our results indicate that α is relatively stable across different settings, with values in the range $[0.1, 0.2]$ providing optimal performance. This stability suggests that the transition function is robust to minor perturbations, making it a reliable component of PCD.

Finally, we explored the effects of varying the base frequencies B and B'. Moderate perturbations (e.g., $B' = 10^4$) were found to be most effective in slowing down the decay rate, as they introduce sufficient variability in low-frequency components without destabilizing the attention mechanism. Extreme values of B', either too small or too large, lead to suboptimal performance, underscoring the importance of carefully calibrating frequency perturbations.

A.3 Details of Case Study: PCD Mitigate PSA

In this case study, we illustrate how PCD can alleviate the PSA phenomenon by examining a specific mismatch between the ground-truth label and the model's prediction. The target label, `ffeae470-29ae-4a8c-9c56-9b97d9edf8ac`, diverges markedly from the model's generated output, `cd501c0360f7`, indicating a notable alignment failure.

To investigate the underlying cause, we first examined the top 18 tokens (by probability) that the model produced during decoding. Specifically, these tokens corresponded to the indices `4484, 1, 15, 25867, 66, 68, 4578, 791, 544, 69, 7047, 98046, 67, 291, 65, 58923, 29069, 16`, which collectively yielded partial strings such as `"cd"` while overlooking crucial prefixes like `"ffe"`. This observation suggests that, despite the ground-truth prefix being comparatively salient in the extended context, the model's unperturbed decoding mechanism favored distractive or irrelevant tokens, thereby failing to align with the target sequence.

Next, we applied PCD to calibrate the model's output distribution by introducing controlled perturbations to low-frequency positional encodings and subsequently contrasting them with the standard logits. By re-ranking the token candidates through this contrastive step, PCD significantly elevated the

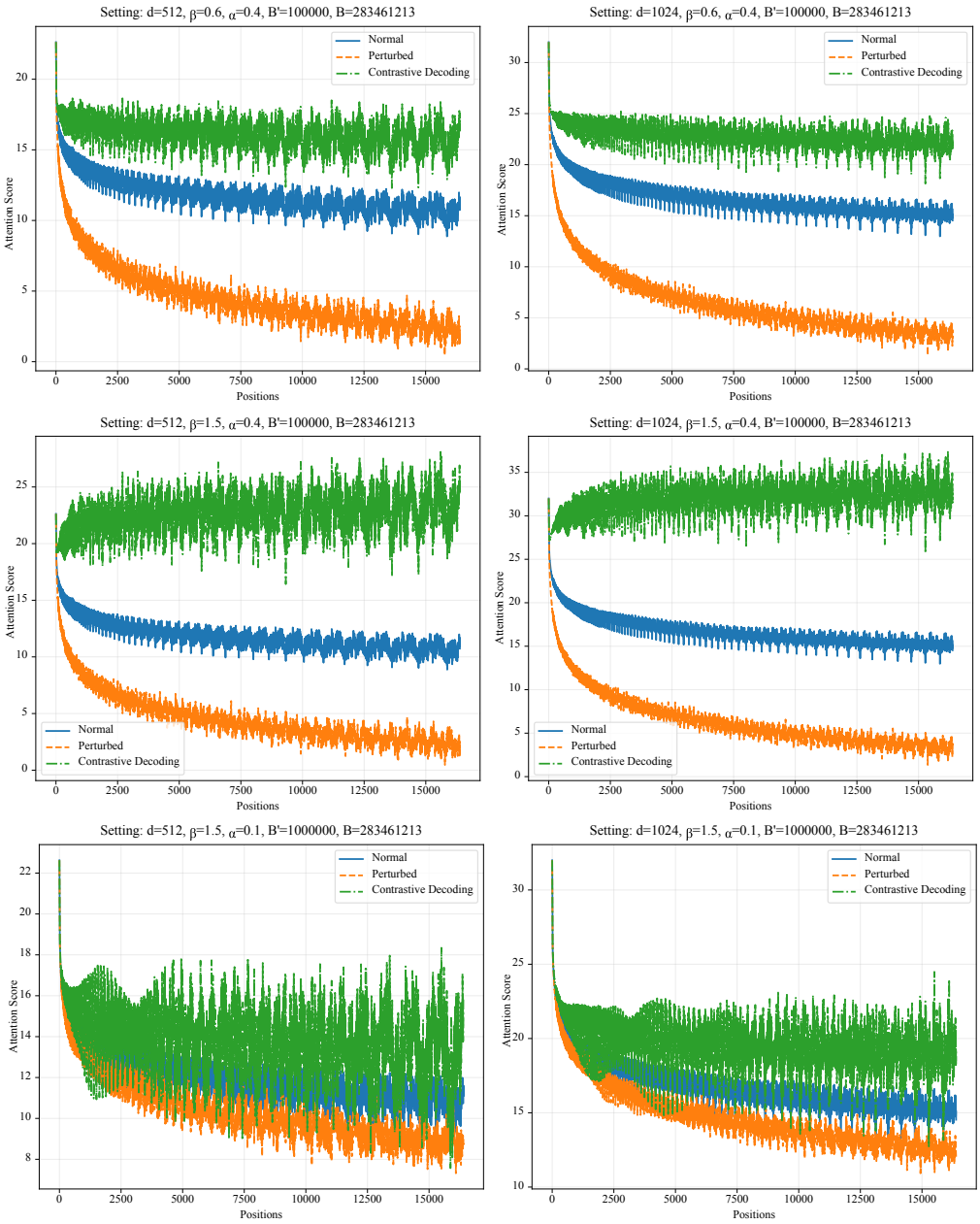


Figure 4: Long-term decay simulation of attention scores under varying hyperparameter settings, including embedding dimension d, contrastive coefficient β, transition function coefficient α, and base frequencies B and B'. The results demonstrate how PCD mitigates attention decay across different configurations, enhancing the model's long-range awareness.

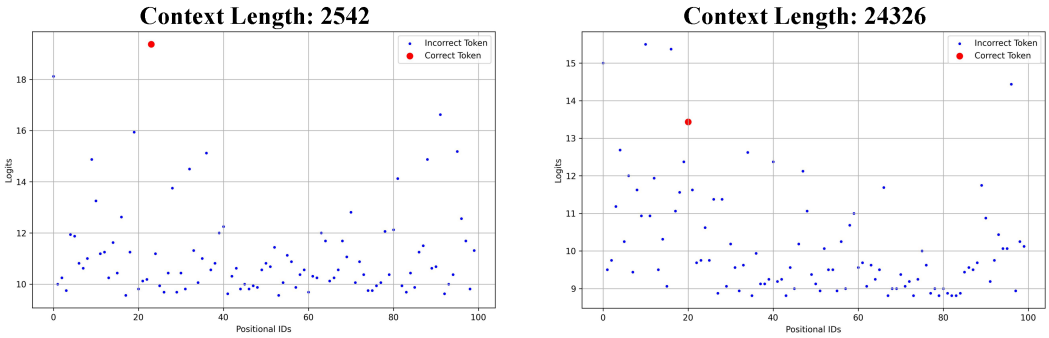


Figure 5: Dynamic visualization of logits as context length increases, illustrating how the logit value of the gold label diminishes relative to other tokens.

position of the correct prefix tokens (e.g., "ffe"). Empirically, this re-ranking suppressed the model's inclination toward incorrect, high-probability distractor tokens and guided the decoding process toward generating the appropriate target label.

The example demonstrate how PCD can effectively mitigate PSA by shifting the model's preference toward contextually relevant tokens. Such recalibration holds particular promise for tasks involving extended contexts or complex sequential outputs, where conventional decoding methods often struggle to discriminate vital information from extraneous content.

A.4 Logits Visualization

To further investigate the PSA phenomenon, we conducted a dynamic visualization of logits as the context length increases. As shown in Fig. 5, the logit value of the gold label gradually diminishes relative to other tokens as the sequence grows longer. This visualization clearly demonstrates how the gold label's prominence is overshadowed by competing logits, particularly in extended contexts. The attenuation of the gold label's salience underscores the challenges faced by long-context models in maintaining focus on critical information, further motivating the need for decoding strategies like PSA (PCD) to mitigate this degradation.

A.5 Related Works

To augment the long-range awareness of Large Language Models (LLMs), research has primarily focused on four strategies: input design, instruction design, model-driven methods, and data-driven methods.

Input design, specifically, involves methods such as segment reranking (Dsouza et al., 2024; Peysakhovich and Lerer, 2023), which aims to mitigate the inherent location bias of transformers by rearranging input segments to prioritize pertinent information during inference. However, this approach suffers from the drawbacks of disrupting semantic coherence and incurring significant computational expenses due to the necessity for multiple inferences. Instruction design (Zhang et al., 2024a; Yu, 2023), employs context recalling, prompting the model to access relevant information before completing a task. Despite its potential to remind the model to retrieve information, context recalling does not inherently enhance long-range awareness. Model-driven methods address long-range awareness by modifying attention mechanisms or positional encodings to diminish attention bias and bolster long-distance awareness. For instance, attention calibration techniques segment attention into local and global encodings to achieve a balanced focus (Zhang et al., 2024b; Hsieh et al., 2024b) However, these modifications necessitate retraining the model, which is not only costly but also poses challenges to maintaining stability and generalizability. Data-driven methods, which involve training on synthetic or multi-document QA datasets, have demonstrated efficacy in improving retrieval accuracy (DATA; An et al., 2024). Nevertheless, the high cost of annotating long-form datasets poses a significant barrier to obtaining high-quality, large-scale training corpora.

Based on the preceding discussion of PSA, it can be observed that the correct tokens are typically positioned very high in the decoding space. The observation holds significant potential for enhancing long-range awareness and improving rank by devising a decoding strategy that promotes the positioning of these tokens higher in the decoding space.

A.6 Rationale for Perturbing Low-Frequency Encodings and Employing Contrastive Decoding

Long-distance awareness is affected by models' positional encodings (PEs) (Su et al., 2024; Press et al., 2022; Chi et al., 2022), which are designed with long-term decay: the farther a token is from the current position, the less relevant its information. The high-frequency encoding is primarily responsible for local modeling, while the low-frequency encoding is responsible for global modeling.

Assuming a function G takes the high-frequency encoding F_h and low-frequency encoding F_l of the positional encoding as inputs, and outputs the model's logits $= G(F_h, F_l)$. Improving the model's global modeling capacity requires enhancing the contribution of the low-frequency signal F_l.

One way involves amplifying the influence of low-frequency encoding, which is accomplished by the standard production of long context training. The other way introduces perturbations to adjust the signal without additional training costs. It offers two options:

(1) Directly perturbing high-frequency encoding to indirectly amplify long-distance awareness: A perturbation ϵ_h is added into high-frequency encoding as $F'_h = F_h + \epsilon_h$. Then the logits expressed as $\mathbf{L} = G(F'_h, F_l)$. However, it is important to emphasize that introducing disturbances to the high-frequency encoding will seriously lead to model collapse.

(2) Initially involving a temporary perturbation of low-frequency encoding, followed by contrastive decoding to reversely amplify long-distance awareness: Perturbing the low-frequency encoding F_l, where a perturbation ϵ_l is added, as $F'_l = F_l + \epsilon_l$. The logits are expressed as $\mathbf{L}_0 = G(F_h, F_l + \epsilon_l)$, and the corrected logits $\mathbf{L}_\alpha$ are computed as $\mathbf{L}_\alpha = (1 + \beta) \cdot \mathbf{L}_0 - \beta \cdot \mathbf{L}_\delta$.

Based on the analysis, directly enhancing long-distance capabilities requires training with long-text data, while perturbing the high-frequency encoding can lead to model collapse. However, as a prospective strategy, perturbing the low-frequency encoding followed by contrastive decoding enhances long-distance awareness without compromising the model's base capabilities.

Sparse-to-Dense: A Free Lunch for Lossless Acceleration of Video Understanding in LLMs

Xuan Zhang[1], **Cunxiao Du**[2*], **Sicheng Yu**[1], **Jiawei Wu**[3], **Fengzhuo Zhang**[3], **Wei Gao**[1], **Qian Liu**[2]
[1]Singapore Management University, [2]Sea AI Lab, [3]National University of Singapore

Abstract

Video Large Language Models (Video-LLMs) suffer from high inference latency in long video processing due to their auto-regressive decoding mechanism, posing challenges for the efficient processing of video sequences that are usually very long. We observe that attention scores in Video-LLMs during decoding exhibit pronounced sparsity, with computational focus concentrated on a small subset of critical tokens. Motivated by this insight, we introduce Sparse-to-Dense (STD), a novel decoding strategy that integrates two distinct modules: a sparse module that rapidly generates speculative tokens using efficient top-K attention, and a dense module that verifies these tokens in parallel via full self-attention. This collaborative approach accelerates Video-LLMs losslessly, effectively offering a free lunch for video understanding. STD is a plug-and-play solution requiring no fine-tuning or architectural changes and achieves up to a 1.94$\times$ wall time speedup while preserving model performance. It enables a seamless conversion of standard Video-LLMs into sparse counterparts, unlocking efficient long-video processing without sacrificing accuracy.

1 Introduction

Recent advances in Video Large Language Models (Video-LLMs), which combine large language models with video understanding, have achieved exceptional performance on tasks like video question answering and captioning (Lin et al., 2024a; Cao et al., 2024; Zhang et al., 2025a). A common practice in Video-LLMs is representing a video as a sequence of image frames, which results in extremely long token sequences that can strain computational resources. For instance, a 1-hour video sampled at 5-second intervals produces 720 frames, which translates to 141,120 visual tokens in VILA (Lin et al., 2024a). These extremely long token sequences cause Video-LLMs to suffer from high inference latency when processing lengthy videos, making real-time applications challenging.

This latency is primarily introduced by the auto-regressive nature of current Video-LLMs, where each new token must attend to all preceding tokens, creating substantial memory and computational challenges.While mechanisms like key-value (KV) caching are employed to store pre-computed key and value tensors and reduce redundant recomputation, frequent access to the cache imposes heavy demands on memory bandwidth due to the growing amount of KV cache with the increasing sequence length. This significantly reduces the throughput of Video-LLMs. A common approach to addressing this problem is KV cache compression (Du et al., 2024b; Chen et al., 2024b; Lin et al., 2024b; Zhang et al., 2025b) or quantization (Su et al., 2025; Hooper et al., 2024; Liu et al., 2024) at test time. However, these methods introduce discrepancies between training and inference, degrading the performance of LLMs.

In this paper, we aim to build a lossless acceleration method designed specifically for Video-LLMs that preserves the exact output distribution of the original model. Although speculative decoding (Leviathan et al., 2023; Chen et al., 2023; Hou et al., 2025) meets this requirement, it usually requires an extra draft model, which is expensive for Video-LLMs. In contrast, we observe that Video-LLMs exhibit a unique structural property, attention sparsity, which can serve as a training-free and plug-and-play draft model. Specifically, retaining only the top-K KV caches in the attention layers preserves the original predictions for approximately 95% of tokens (empirically verified), suggesting that most attention heads contribute minimally to the final output. Motivated by this observation, we introduce a novel decoding method called Sparse-to-Dense (STD), which leverages the sparse structure of Video-LLMs as its

*Corresponding author: ducx@sea.com

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 734–742
July 27 - August 1, 2025

draft model. This design eliminates the need for an extra trained draft model, making STD a plug-and-play solution. We refer to the original Video-LLM as the dense model because it decodes using the full KV cache, whereas the model with top-K attention is termed the sparse model. Both models share identical architectures, differing only in how they compute attention. Therefore, we do not need additional GPU memory to store the sparse model, nor does it require any extra training. The top-K attention in the sparse model boosts decoding speed while sacrificing some token quality, whereas the dense model is slower but guarantees accuracy. We use the sparse model to auto-regressively draft the next γ tokens, while the dense model verifies them in parallel. This approach avoids redundant full KV cache memory and ensures the outputs exactly match those of the original Video-LLM.

We conduct experiments on representative Video-LLMs including LLaVA-OneVision (Li et al., 2024a) and Qwen2-VL (Wang et al., 2024), evaluating them on video understanding benchmarks like MLVU (Zhou et al., 2024) and VideoMME (Fu et al., 2024). Experiment results show that our STD, serving as a tuning-free, plug-and-play solution, achieves up to a 1.94× acceleration of video input processing without any performance degradation. It is immediately deployable, requiring only 20 lines of code to transform an original Video-LLM into a sparse Video-LLM, and it does not require any extra training to deploy the draft model.

2 Observation

In this section, we investigate the disparity in decoded tokens between two configurations of Video-LLMs: 1) sparse top-K KV cache: utilizing only the top-K KV caches based on the highest attention weights; and 2) dense full KV cache: employing the complete set of KV caches. We conduct experiments using the Qwen2-VL-7B (Wang et al., 2024) model on randomly selected samples from MLVU (Zhou et al., 2024), and Video-MME (Fu et al., 2024) datasets. We evaluate the next-token prediction accuracy of the model when employing sparse attention with top-K KV caches. Our findings indicate that the model with sparse attention maintains an average token prediction accuracy exceeding 95%. This high accuracy suggests that for the majority of decoded tokens, only the top-K KV caches are necessary. However, it is important to note that the 95% accuracy is measured per individual token and does not accumulate across multiple tokens. For instance, the accuracy of correctly predicting five consecutive tokens drops to approximately $(95\%)^5 \approx 77\%$.

3 Method

In this section, we present Sparse-to-Dense (STD), a method designed to achieve lossless acceleration for Video-LLMs. We refer to the original model $\mathcal{M}$ as the dense model, as it requires the full KV cache during decoding, while the sparse model $\mathcal{M}_s$ uses sparse attention. Although $\mathcal{M}_s$ is faster, it is somewhat less accurate. Unlike traditional speculative decoding, which relies on an additional draft model, our approach leverages $\mathcal{M}_s$ with the same parameters as $\mathcal{M}$. The only difference is that $\mathcal{M}_s$ loads a reduced KV cache to perform sparse attention, eliminating the need for extra GPU memory to store another model's parameters. In the following subsections, we will detail the decoding procedure and the design of the sparse model.

3.1 Decoding Procedures

In our STD, the sparse model $\mathcal{M}_s$ functions as a draft model to propose potential next γ tokens, while the dense model $\mathcal{M}$ verifies them to derive the final output sequence. Given an input sequence $\{x_0, \cdots, x_{m-1}\}$, consisting of visual and textual tokens , the sparse model $\mathcal{M}_s$ auto-regressively generates γ subsequent token candidates $\{x_m, \cdots, x_{m+\gamma-1}\}$. Because the tokens proposed by the sparse model $\mathcal{M}_s$ might not align with those predicted by the dense model $\mathcal{M}$, it requires the verification of $\mathcal{M}$. The dense model $\mathcal{M}$ verifies all γ proposed tokens in parallel, requiring only a single I/O operation for the full KV cache. Thus, this verification procedure accelerates the process compared with the auto-regressive decoding of $\mathcal{M}$ itself, where each token requires a separate I/O operation. During the verification, $\mathcal{M}$ identifies the first n tokens that align with its predictions, where $0 \leq n \leq \gamma$, and additionally provides a bonus token $\hat{x}_{n+m}$ for free. The verified sequence $\{x_m, \cdots, x_{m+n-1}, \hat{x}_{n+m}\}$ is then appended to the input sequence $\{x_0, \cdots, x_{m-1}\}$ to form the context for the next round of proposal and verification.

3.2 Model with Sparse Attention

Next, we introduce the design of our sparse model $\mathcal{M}_s$. Empirical observations in Section 2 indicate

that during most decoding steps, attention scores are predominantly concentrated on a small subset of KV caches, a pattern we term *sparse* attention (also known as top-K attention (Lou et al., 2024)). Only a small fraction of tokens require more evenly distributed *dense* attention. This insight motivates a strategy to selectively apply sparse attention for the majority of tokens and resort to dense attention only when necessary, reducing the I/O overhead of accessing the full KV cache, and thereby improving decoding speed.

Since the number of visual tokens is typically much larger than the number of textual tokens ($m_v \gg m_t$), with m_v often exceeding 10,000 while m_t are usually around 100, our primary focus is on reducing the size of the visual KV cache. To achieve this, we leverage the attention patterns of the textual tokens X_t to identify and select the most relevant KV caches from the visual tokens. Specifically, we analyze the allocation of attention scores when processing the textual tokens $X_t = \{x_{m_v}, \cdots, x_{m-1}\}$ (i.e., the last m_t tokens in the input sequence) to identify which KV pairs of the visual tokens X_v contribute more during the prefilling stage. For each layer l, we calculate the average attention scores directed toward the visual tokens X_v for textual tokens X_t. We then retain only the top-K KV pairs of visual tokens with the highest attention scores. To balance performance and efficiency, we determine the retained K KV caches only during the prefilling stage and avoid the computation-demand dynamic selections in the decoding stage. The selected visual tokens can vary across different layers and attention heads, reflecting the distinct focus of each layer and head in processing the input. The selection of the KV cache of layer l can be formalized as

$$\text{Cache}_s[l] = \text{argTopK}_{x \in X_v}\left(\frac{1}{m_t}\sum_{\hat{x} \in X_t} A_l(\hat{x}, x)\right),$$

where argTopK$(\cdot)$ is an operation that selects the top-K elements indices with the highest values from a given set, k is a predefined hyper-parameter, and $A_l(\hat{x}, x)$ represents the attention score from token $\hat{x}$ to token x in layer l. For models utilizing Grouped Query Attention (GQA) (Ainslie et al., 2023), where the number of query heads equals the number of groups multiplied by the number of KV heads, we directly sum the attention scores within each group to select the top-K KV caches for this head. The KV cache selection operates at the granularity of individual KV heads, allowing each layer or head to retain a distinct subset of caches based on its specific requirements.

3.3 I/O complexity analysis.

In the decoding phase, the I/O complexity of our Sparse-to-Dense decoding method can be analyzed as follows. For the sparse model $\mathcal{M}_s$, which speculatively proposes γ subsequent tokens, the I/O cost involves accessing the selected K visual KV caches and all m_t textual KV caches. Thus, the total I/O for the sparse model is given by: $\text{I/O}_{\text{sparse}} = \gamma \times (K + m_t)$. For the dense model $\mathcal{M}$, which verifies the proposed tokens in parallel, the I/O cost includes accessing the full KV caches of all visual and textual tokens, resulting in: $\text{I/O}_{\text{dense}} = m_v + m_t$. The total I/O for Sparse-to-Dense decoding is therefore: $\text{I/O}_{\text{total}} = \gamma \times (K + m_t) + (m_v + m_t)$, and the average I/O per token is

$$\text{I/O}_{\text{average}} = \frac{\text{I/O}_{\text{total}}}{\alpha \times \gamma} = \frac{\gamma \times (K + m_t) + m_v + m_t}{\alpha \times \gamma},$$

where α ratio of the number of accepted tokens among all proposed tokens. In contrast, the average I/O complexity of vanilla decoding, where each token is generated using full attention, is given by: $\text{I/O}_{\text{average}}^{\text{vanilla}} = m_v + m_t$. When α is sufficiently large, i.e., $\alpha > (K + m_t)/(m_v + m_t) + \gamma^{-1}$, the average I/O per token in our method becomes considerably lower, resulting in improved decoding efficiency. Intuitively, we hope that the ratio between the numbers of the accepted tokens and all proposed tokens is larger than the ratio between the numbers of retrained KV pairs and the full KV cache. This can be achieved due to the concentration behavior of attention scores in Section 2. The empirical superiority of our method in the next section verifies this inequality in the realistic setting.

4 Experiment

Baselines. To evaluate the effectiveness of our proposed Sparse-to-Dense decoding, we compare it against the following baselines: 1) Layerskip (Elhoushi et al., 2024): This method utilizes a model with an layer-level early exit mechanism to propose draft tokens. This baseline is inspired by the work of Elhoushi et al. on text-only LLMs, and originally requires additional training. For a fair comparison with our method, we adapt it to Video-LLMs in a tuning-free manner. 2) Streaming (Chen et al., 2024a): This method employs a model with streaming attention (Xiao et al., 2023) to propose

Methods	MLVU		VideoMME-s		VideoMME-m		VideoMME-l	
	Acc. (%)	Speedup	Acc. (%)	Speedup	Acc. (%)	Speedup	Acc. (%)	Speedup
LLaVA-OneVision-7B								
LayerSkip	10.0	0.47×	5.6	0.33×	8.1	0.46×	4.8	0.44×
Streaming	34.7	1.34×	36.4	1.38×	41.0	1.51×	36.2	1.45×
STD (ours)	**47.8**	**1.72**×	**51.8**	**1.82**×	**52.1**	**1.83**×	**52.9**	**1.59**×
Qwen2-VL-7B-Instruct								
LayerSkip	5.2	0.63×	3.7	0.59×	4.9	0.55×	5.7	0.55×
Streaming	53.9	1.61×	52.9	1.32×	59.2	1.36×	59.6	1.36×
STD (ours)	**66.1**	**1.94**×	**71.8**	**1.71**×	**73.4**	**1.62**×	**81.8**	**1.70**×

Table 1: Comparisons of the acceptance rate (Acc.) and wall time speedup of STD and previous draft models. **Bold** denotes the best method. Since all the methods are lossless, we do not report the evaluation of the generated contents.

draft tokens. Similar to LayerSkip, this baseline is derived from the work of Chen et al. on text-only LLMs. To ensure comparability with our approach, we extend its implementation to Video-LLMs.

Datasets and evaluation metrics. We evaluate Sparse-to-Dense on two widely adopted benchmarks: MLVU (Zhou et al., 2024) and VideoMME (Fu et al., 2024). MLVU is specifically designed for long-duration videos, while VideoMME encompasses short, medium, and long-duration videos, providing a comprehensive assessment across various video lengths. For our evaluation, we adhere to the protocols established in previous works on speculative decoding. We report two primary metrics: ***acceptance rate*** of the draft tokens and wall time ***speedup***.

Implementation Details. Our experiments are conducted using widely adopted state-of-the-art Video-LLMs, specifically LLaVA-OneVision (7B) (Li et al., 2024a) and Qwen2-VL (7B) (Wang et al., 2024). We prompt the Video-LLMs to generate chain-of-thought (Wei et al., 2022) responses to enhance their performance. We set the sum of the textual token count m_t and the selected visual KV cache count K to 1024, with a batch size of 8. The number of tokens verified by the dense model $\mathcal{M}_d$ is fixed at $\gamma = 9$. The ablation of hyperparameters can be found in Appendix Section C. Our framework is implemented based on Hugging Face's Transformers library. All experiments are conducted on NVIDIA A100 GPUs with 80 GB of memory, and are repeated three times with different random seeds, and the average results are reported.

Main Results Table 1 summarizes the performance across various reasoning tasks. We have the following findings: 1) *The draft model based on LayerSkip performs worse than that utilizing sparse attention* (e.g., Streaming and STD). The primary reason for this discrepancy is that LayerSkip causes a substantial distributional shift between the draft model and the target model, leading to a low acceptance rate. Although the draft model with layer skipping runs considerably faster than the sparse attention counterparts, this advantage is insufficient to compensate for the overall wall-time speedup loss introduced by layer skipping. 2) *Draft models based on sparse attention generally provide more wall time speedup.* Whether in STD or Streaming, we observe a consistently high acceptance rate. This indicates that, for most of the time, the target model does not require the full KV cache but only a sparsely selected subset cache. However, it is important to note that since LLMs perform autoregressive decoding, an incorrect token can propagate errors to subsequent tokens. Thus verification with the full KV cache is essential. 3) *Our model outperforms the streaming-based draft model*, achieving 62.2% in acceptance length and 1.74× in wall-time speedup on average. This advantage stems from our method's ability to leverage the unique characteristics of Video-LLMs to select important KV cache. As observed in section 2, text-guided video cache selection effectively identifies and retains the most critical cache elements.

5 Conclusion

We introduce STD, a training-free, plug-and-play decoding method that employs sparse top-K attention as the draft model in speculative decoding while leveraging full attention for verification in parallel, ensuring lossless acceleration. Extensive experiments demonstrate that STD significantly outperforms strong baselines that use LayerSkip and Streaming as the draft models. Overall, STD achieves up to a 1.94× walltime speedup while maintaining identical output quality. In the future, we hope to extend our work to accelerate long

CoT Video-LLMs such as QvQ (QwenLM Team, 2024).

Limitation

A notable limitation of our current approach is that all KV caches are still stored in GPU memory (i.e., HBM). While HBM provides the high bandwidth necessary for fast computations, its capacity is inherently limited, which poses a significant bottleneck during inference—especially as model sizes and sequence lengths increase. The limited HBM capacity may lead to restricted batch size.

In the future, a promising solution to this challenge is to offload portions of the KV caches to CPU memory. Although CPU memory typically has lower bandwidth compared to HBM, it offers substantially larger capacity. By developing efficient data transfer and caching strategies, it may be possible to mitigate the HBM bottleneck without sacrificing inference accuracy, thereby enabling more scalable and efficient processing for large Video-LLMs.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*.

Jianjian Cao, Peng Ye, Shengze Li, Chong Yu, Yansong Tang, Jiwen Lu, and Tao Chen. 2024. Madtp: Multimodal alignment-guided dynamic token pruning for accelerating vision-language transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15710–15719.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Jian Chen, Vashisth Tiwari, Ranajoy Sadhukhan, Zhuoming Chen, Jinyuan Shi, Ian En-Hsu Yen, and Beidi Chen. 2024a. Magicdec: Breaking the latency-throughput tradeoff for long context generation with speculative decoding. *arXiv preprint arXiv:2408.11049*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024b. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. *arXiv preprint arXiv:2403.06764*.

Cunxiao Du, Jing Jiang, Xu Yuanchen, Jiawei Wu, Sicheng Yu, Yongqi Li, Shenggui Li, Kai Xu, Liqiang Nie, Zhaopeng Tu, et al. 2024a. Glide with a cape: A low-hassle method to accelerate speculative decoding. *arXiv preprint arXiv:2402.02082*.

Cunxiao Du, Hao Zhou, Zhaopeng Tu, and Jing Jiang. 2024b. Revisiting the markov property for machine translation. *arXiv preprint arXiv:2402.02084*.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, et al. 2024. Layer skip: Enabling early exit inference and self-speculative decoding. *arXiv preprint arXiv:2404.16710*.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. 2024. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*.

Mukul Gagrani, Raghavv Goel, Wonseok Jeon, Junyoung Park, Mingu Lee, and Christopher Lott. 2024. On speculative decoding for multimodal large language models. *arXiv preprint arXiv:2404.08856*.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. 2024. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *arXiv preprint arXiv:2401.18079*.

Yunlong Hou, Fengzhuo Zhang, Cunxiao Du, Xuan Zhang, Jiachun Pan, Tianyu Pang, Chao Du, Vincent YF Tan, and Zhuoran Yang. 2025. Banditspec: Adaptive speculative decoding via bandit algorithms. *arXiv preprint arXiv:2505.15141*.

Zhengmian Hu and Heng Huang. Accelerated speculative sampling based on tree monte carlo. In *Forty-first International Conference on Machine Learning*.

Doohyuk Jang, Sihwan Park, June Yong Yang, Yeonsung Jung, Jihun Yun, Souvik Kundu, Sung-Yub Kim, and Eunho Yang. 2024. Lantern: Accelerating visual autoregressive models with relaxed speculative decoding. *arXiv preprint arXiv:2410.03355*.

Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. 2024. Vidcompress: Memory-enhanced temporal compression for video understanding in large language models. *arXiv preprint arXiv:2410.11417*.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024b. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. 2024a. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699.

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. 2024b. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. *arXiv preprint arXiv:2405.05803*.

Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Alvin Cheung, Zhijie Deng, Ion Stoica, and Hao Zhang. 2023. Online speculative decoding. *arXiv preprint arXiv:2310.07177*.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*.

Chao Lou, Zixia Jia, Zilong Zheng, and Kewei Tu. 2024. Sparser is faster and less is more: Efficient sparse attention for long-range transformers. *arXiv preprint arXiv:2406.16747*.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, et al. 2024. Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pages 932–949.

QwenLM Team. 2024. Qvq-72b preview. https://qwenlm.github.io/blog/qvq-72b-preview/.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. 2024. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*.

Dingjie Song, Wenjun Wang, Shunian Chen, Xidong Wang, Michael Guan, and Benyou Wang. 2024. Less is more: A simple yet effective token reduction method for efficient multi-modal llms. *arXiv preprint arXiv:2409.10994*.

Zunhai Su, Wang Shen, Linge Li, Zhe Chen, Hanyu Wei, Huangqi Yu, and Kehong Yuan. 2025. Akvq-vl: Attention-aware kv cache adaptive 2-bit quantization for vision-language models. *arXiv preprint arXiv:2501.15021*.

Ziteng Sun, Jae Hun Ro, Ahmad Beirami, and Ananda Theertha Suresh. 2024a. Optimal block-level draft verification for accelerating speculative decoding. *arXiv preprint arXiv:2403.10444*.

Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix Yu. 2024b. Spectr: Fast speculative decoding via optimal transport. *Advances in Neural Information Processing Systems*, 36.

Ruslan Svirschevski, Avner May, Zhuoming Chen, Beidi Chen, Zhihao Jia, and Max Ryabinin. 2024. Specexec: Massively parallel speculative decoding for interactive llm inference on consumer devices. *arXiv preprint arXiv:2406.02532*.

Yao Teng, Han Shi, Xian Liu, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. 2024. Accelerating auto-regressive text-to-image generation with training-free speculative jacobi decoding. *arXiv preprint arXiv:2410.01699*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yuxin Wen, Qingqing Cao, Qichen Fu, Sachin Mehta, and Mahyar Najibi. 2024. Efficient vision-language models by summarizing visual tokens into compact registers. *arXiv preprint arXiv:2410.14072*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Linli Yao, Lei Li, Shuhuai Ren, Lean Wang, Yuanxin Liu, Xu Sun, and Lu Hou. 2024. Deco: Decoupling token compression from semantic abstraction in multimodal large language models. *arXiv preprint arXiv:2405.20985*.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. 2025a. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*.

Xuan Zhang, Fengzhuo Zhang, Cunxiao Du, Chao Du, Tianyu Pang, Wei Gao, and Min Lin. 2025b. Light-transfer: Your long-context llm is secretly a hybrid model with effortless adaptation. In *Workshop on Reasoning and Planning for Large Language Models*.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2024. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*.

A Preliminary

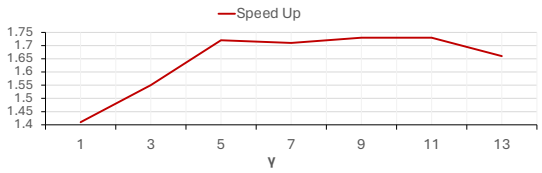
Speculative Decoding We first formalize our notation and provide a brief overview of the speculative decoding in autoregressive LLMs, which is the key background knowledge for our method. We represent the input sequence for a Video-LLM as a combination of visual tokens and textual tokens. Specifically, the visual tokens are denoted as $X_v = \{x_0, \cdots, x_{m_v-1}\}$, and the textual prompt is denoted as $X_t = \{x_{m_v}, \cdots, x_{m-1}\}$. Here, m_v is the number of visual tokens, m_t is the number of textual tokens, and the total input sequence length is $m = m_v + m_t$. The key and value cache for token x_i are represented by K_{x_i} and V_{x_i}, respectively.

Inference of Auto-regressive Models. The inference stage of auto-regressive models, e.g., Video-LLMs, can be divided into two stages: 1) *prefilling*: The video LLM processes the input sequence, which includes both visual tokens X_v and textual tokens X_t, in an autoregressive and parallel manner. For each token x_i in the combined input $\{X_v, X_t\}$, the model computes and stores the corresponding KV cache entries. This stage effectively encodes the input sequence and prepares the model for generating a response. The output of this stage is the first token x_m of the model's response. 2) *decoding*: After prefilling, the model enters the decoding phase, generating output tokens sequentially. At each decoding step $j = m+1, m+2, \cdots$, the video LLM generates a new token x_j based on the KV cache from all prior tokens. After generating, the KV cache is updated with each newly generated token. This process continues iteratively until a stopping criterion is met, such as reaching an end-of-sequence token or hitting a maximum token limit.

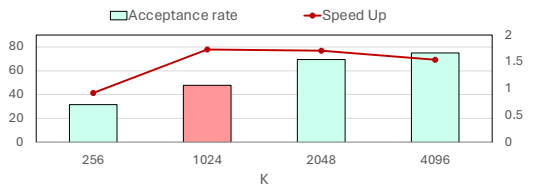
B Related Works

Sparse Attention in MLLMs Normally, an image or a video frame is represented as a large number of tokens in MLLMs, e.g., 196 visual tokens per image in VILA (Lin et al., 2024a), which significantly impacts the computational and storage during model training and inference. Visual token compression aims to reduce the number of visual tokens to address it directly. The majority of visual token compression methods either train from scratch or perform additional training based on existing models. For example, some image-based MLLMs rely on vision-language alignment (Cao et al., 2024; Yao et al., 2024; Song et al., 2024) or aggressively removing all visual tokens after a certain layer (Wen et al., 2024), while methods designed for video-based MLLMs consider the unique characteristics of video, such as employing memory mechanisms (Lan et al., 2024) or compressing tokens along spatial and temporal dimensions sequentially (Shen et al., 2024). A smaller portion of works study the test-time (training free) visual token compression for accelerating the inference procedure. FastV (Chen et al., 2024b) performs pruning by analyzing the attention pattern from shallow layers and deep layers, while another approach directly applies full visual token removal during the inference stage (Lin et al., 2024b). In our method, STD, the design of the drafter model is related to training-free visual token compression techniques. However, these previous methods inevitably impact the original model's performance. In contrast, we propose to utilize visual token compression as a drafter model to achieve lossless inference acceleration.

Speculative Decoding Speculative decoding is proposed by (Leviathan et al., 2023) and (Chen et al., 2023) to accelerate the inference of LLMs, where the throughput of LLMs is improved $2 \sim 3$ times without sacrificing the performance. The algorithm consists of two stages: drafting and verification. The drafting stage adopts a small model (drafter) to generate a long sequence of possible future tokens swiftly, while the verification stage accepts a part of the tokens predicted in the drafting stage in a token-by-tone manner. The follow-up improves the speculative decoding from these two perspectives. Specinfer (Miao et al., 2024), Eagle (Li et al., 2024b) and Medusa (Cai et al., 2024) propose to train a drafter to generate tokens with a tree structure, and the verification is conducted on the tree in a branch-by-branch manner. Hu and Huang (Hu and Huang) also organize the draft tokens as a tree, but they verify the tokens in a branch as a whole. Glide (Du et al., 2024a) generates draft tokens as an unbalanced tree, which alleviates the burden of the drafter while achieving significant acceleration. SpecTr (Sun et al., 2024b) views speculative decoding from the optimal transport view and proposes to verify a batch of draft tokens jointly. They show that the proposed algorithm is optimal up to a multiplicative factor. Sun et al. (Sun et al., 2024a) boot the acceleration by a joint verification



(a)



(b)

Figure 1: Effect of K and γ on MLVU using LLaVA-OneVision-7B.

of a single draft trajectory. Instead of using a token-by-token manner, they accept the draft sentences as a whole. Lie et al. (Liu et al., 2023) proposes to update the parameters of drafters in an online manner, which is shown to be effective in various applications. MagicDec (Chen et al., 2024a) analyzes the speculative decoding in the long-context setting with an emphasis on the FLOPS and memory. SpecExec (Svirschevski et al., 2024) focuses on a special setting where the LLMs are offloading their parameters. Several works (Gagrani et al., 2024; Jang et al., 2024; Teng et al., 2024) study the speculative decoding of MLLMs. However, they focus either on the image understanding problem or the image generation problem. In contrast, our work is the first to study video generation acceleration via speculative decoding.

C Ablation Stuidies

We also conducted additional experiments to analyze the impact of hyperparameters (γ and K) on model performance. As shown in Figure 1a, we can see that as $gamma$ increases, the speed up gradually improves. This improvement is because the sparse model makes accurate predictions, which allows the computational overhead to be spread out over more tokens. However, when $gamma$ reaches 13, the speed up starts to decline because the model's accuracy in correctly predicting 13 consecutive tokens is insufficient. At the same time, as shown in Figure 1b, when K is small, the acceptance rate is low, resulting in a lower speed up. In contrast, when K is large, the sparse model is not as fast, which also leads to a reduced speed-up.

Revisiting Uncertainty Quantification Evaluation in Language Models: Spurious Interactions with Response Length Bias Results

Andrea Santilli[*†] Sapienza University of Rome
Adam Goliński[*] Apple
Michael Kirchhof Apple
Federico Danieli Apple
Arno Blaas Apple
Miao Xiong[†] National University of Singapore
Luca Zappella Apple
Sinead Williamson Apple

Abstract

Uncertainty Quantification (UQ) in Language Models (LMs) is key to improving their safety and reliability. Evaluations often use metrics like AUROC to assess how well *UQ methods* (e.g., negative sequence probabilities) correlate with task *correctness functions* (e.g., ROUGE-L). We show that mutual biases-when both *UQ methods* and *correctness functions* are biased by the same factors-systematically distort evaluation. First, we formally prove that any mutual bias non-randomly skews AUROC rankings, compromising benchmark integrity. Second, we confirm this happens empirically by testing 7 widely used *correctness functions*, from lexical-based and embedding-based metrics to LM-as-a-judge approaches, across 4 datasets $\times$ 4 models $\times$ 8 *UQ methods*. Our analysis shows that length biases in *correctness functions* distort UQ assessments by interacting with length biases in *UQ methods*. We identify LM-as-a-judge methods as the least length-biased, offering a promising path for a fairer UQ evaluation.

1 Introduction

Language Models (LMs) excel at natural language generation but often produce factually incorrect outputs, or "hallucinations" (Guerreiro et al., 2023; Huang et al., 2025). These hallucinations are typically associated with high uncertainty about the correct output (Xiao and Wang, 2021), leading to the emergence of *Uncertainty Quantification (UQ) methods* as a compelling approach to detect errors (Farquhar et al., 2024; Baan et al., 2023). A fundamental challenge in evaluating *UQ methods* is the lack of ground truth uncertainty labels. Consequently, benchmarks commonly rely on *UQ performance metrics* such as AUROC, assessing how effectively *UQ methods* distinguish correct from incorrect outputs as determined by a *correctness function*. Thus, the accuracy and reliability of UQ evaluations inherently depend on the quality of the correctness assessments.

In this paper, we critically analyze how errors and biases in *correctness functions* impact *UQ performance metrics*. First, we provide a formal analysis showing that: i) if errors in the *correctness function* are random and independent from the *UQ method*, AUROC is noisy but unbiased; ii) conversely, if there exists a *mutual bias*—i.e. if the *correctness function* errors correlate systematically with the uncertainty scores—then AUROC rankings are inherently skewed. Our formal results demonstrate that **any mutual bias** introduces systematic distortions into AUROC evaluations, artificially advantaging certain methods and fundamentally undermining the reliability of benchmarks.

We confirm this happens empirically by benchmarking 7 widely-used *correctness functions*, including lexical-based metrics (e.g., ROUGE metrics (Lin, 2004)), embedding-based metrics (e.g., BERTScore (Zhang et al., 2020)), and LM-as-a-judge approaches (Zheng et al., 2024) across 4 datasets $\times$ 4 models $\times$ 8 *UQ methods*. We reveal two key issues: (i) the *correctness function* choice significantly impacts UQ results and (ii) widely used lexical-based and embedding-based *correctness functions* (Farquhar et al., 2024; Fadeeva et al., 2023) introduce systematic biases that distort the perceived effectiveness of certain *UQ methods*.

A human evaluation of 450 LM samplesreveals that this bias stems, at least in part, from the mutual dependence of certain *UQ methods* and *correctness functions* on the output length. Building on this, we identify *correctness functions* that mitigate bias by avoiding such confounding, finding LM-as-a-judge approaches best suited for UQ evaluation and most aligned with human judgment. Overall, this study highlights widespread pitfalls in UQ evaluation and charts a path toward a more reliable evaluation protocol.

[*]Equal contribution
[†]Work done during an internship at Apple.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 743–759
July 27 - August 1, 2025

2 Evaluating uncertainty

In this section, we review common *UQ methods*, *correctness functions* and *UQ performance metrics*.

2.1 *UQ methods*

Given an input x to an LM, which generates an output sequence $\hat{y}$, a *UQ method* estimates a measure of the model's uncertainty about $\hat{y}$, denoted as $\hat{g}(\hat{y}, x)$. These methods can be broadly categorized into three types: (i) single-sample, (ii) multi-sample, and (iii) learned. One simple single-sample approach is negative sequence probability,

$$\hat{g}(\hat{y}, x) = -\hat{p}(\hat{y}|x) = -\prod_{i=1}^{L} \hat{p}(\hat{y}_i|\hat{y}_{<i}, x), \quad (1)$$

where L is the length of the generated answer and $\hat{p}$ the output probabilities assigned by the model. Note that in Eq. 1, $\hat{g}$ increases with L. Multiple-sample approaches derive uncertainty scores by sampling multiple responses for the same input x and measuring a metric (e.g., variance) across samples. Notably, several *UQ methods* in this second group, like Naïve Entropy and Semantic Entropy (Farquhar et al., 2024), use Eq. 1 to compute the probability of a sequence of generated tokens. Lastly, learned methods train a binary classifier via supervised learning on a correctness-labeled dataset (Kadavath et al., 2022). We refer readers to App. A for more details on each family.

2.2 *Correctness functions*

Correctness functions $\hat{h}(\hat{y}, x, y)$ compare a generated answer $\hat{y}$ to a reference answer y to estimate a correctness score, and can be categorized as lexical-based, embedding-based, or LM-as-a-judge.

Lexical-based *correctness functions*, such as SQuAD (Rajpurkar et al., 2016) and ROUGE variants (Lin, 2004), are based on lexical overlap between $\hat{y}$ and y. While limitations of these metrics have been studied in areas like summarization and Question Answering (QA) (Guo and Vosoughi, 2023; Chen et al., 2019; Cohan and Goharian, 2016; Fabbri et al., 2021; Reiter and Belz, 2009), their impact on UQ evaluation remains largely unexplored.

Embedding-based *correctness functions*, such as BERTScore (Zhang et al., 2020) and Sentence-BERT cosine similarity (Reimers and Gurevych, 2019), assess similarity by encoding both $\hat{y}$ and y using a language model, typically BERT-based.

Correctness function	Used in UQ eval protocol	Threshold t
ROUGE-1 (F1)	Aichberger et al. (2025)	$0.1 - 1.0$
ROUGE-L (F1)	Fadeeva et al. (2023); Kuhn et al. (2023)	0.5
	Duan et al. (2024); Chen et al. (2024a)	0.5
	Qiu and Miikkulainen (2024)	$0.1 - 1.0$
	Aichberger et al. (2025)	$0.1 - 1.0$
SQuAD (F1)	Farquhar et al. (2024)	0.3
BERTScore (F1)	Fadeeva et al. (2023)	N/A
SentenceBERT	Chen et al. (2024a)	0.9
AlignScore	Vashurin et al. (2025)	0.5
LM-as-a-judge (Prompt)	Farquhar et al. (2024)	N/A

Table 1: *Correctness functions* used in UQ evals.

LM-as-a-judge *correctness functions* evaluate correctness by using another LM to judge the accuracy of $\hat{y}$ against y. Examples include AlignScore (Zha et al., 2023), which uses a specifically trained LM, and prompt-based variants of LM-as-a-judge (Zheng et al., 2024).

Table 1 summarizes common *correctness functions* used in recent UQ papers. AUROC requires binary labels, so a certain threshold t is typically applied to binarize continuous correctness scores. Some *correctness functions* are inherently binary (e.g., LM-as-a-judge), and some *UQ performance metrics* do not require binarization (Fadeeva et al., 2023). This variety in UQ eval protocols raises questions about which combination to trust. App. B offers a broader view on each metric.

2.3 *UQ performance metrics*

The utility of *UQ methods* is typically assessed using a *UQ performance metric* that quantifies how well uncertainty estimates (§2.1) correlate with correctness. Among the various *UQ performance metrics* available in the literature (Malinin and Gales, 2020; Fadeeva et al., 2023), we focus on the Area Under the Receiver Operating Characteristic curve (AUROC) due to its widespread use in UQ benchmarks (Farquhar et al., 2024; Chen et al., 2024a).

Let $\hat{g}_i \equiv \hat{g}(\hat{y}_i, x_i)$ be the uncertainty score assigned by some *UQ method* to the i-th data sample, and let h_i be a binary label denoting (ground truth) correctness of that data sample ($h_i = 1$ if correct, $h_i = 0$ if incorrect). AUROC can be written as

$$\text{AUROC} = P\left(\hat{g}_i < \hat{g}_j \mid h_i = 1,\ h_j = 0\right), \quad (2)$$

i.e., the probability that a randomly chosen *correct* data sample receives a lower uncertainty score than a randomly chosen *incorrect* data sample.

2.4 Mutual biases in *UQ performance metric*

In practice, we estimate h_i using a *correctness function* $\hat{h}_i \equiv \hat{h}(\hat{y}_i, x_i, y_i)$ from §2.2. This means that

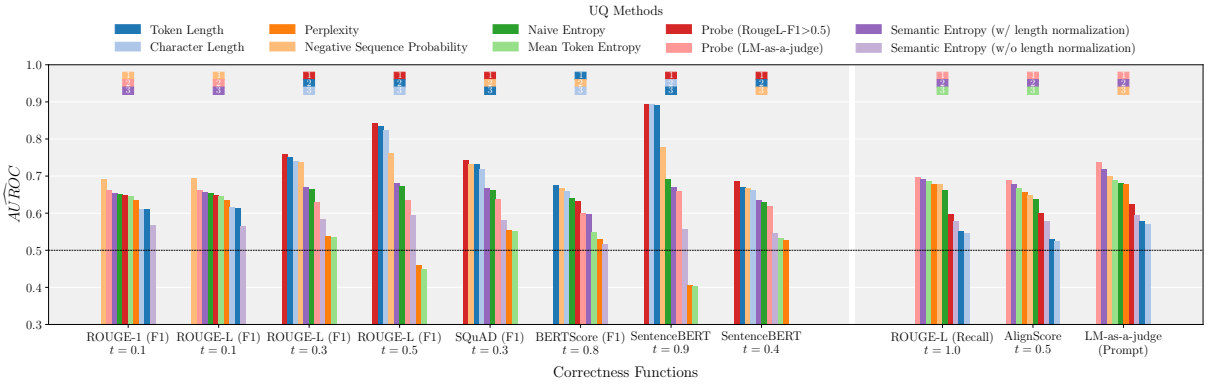


Figure 1: $\widehat{AUROC}$ of various *UQ methods* across *correctness functions* averaged over models and datasets. The ranking of *UQ methods* (top row) changes across *correctness functions*, raising questions about which one to trust.

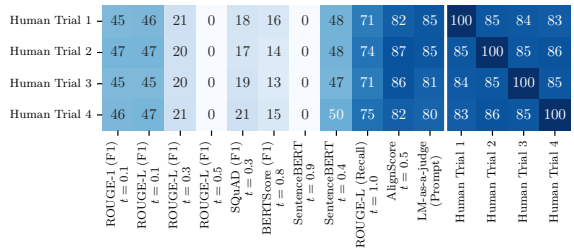


Figure 2: Cohen Kappa agreement rates between annotators and *correctness functions*. Per dataset: Fig. 6.

AUROC is not computed on ground-truth labels, but rather on a potentially biased surrogate:

$$\widehat{\text{AUROC}} = P\left(\hat{g}_i < \hat{g}_j \mid \hat{h}_i = 1,\ \hat{h}_j = 0\right). \quad (3)$$

Eq. 3 highlights a key challenge: the measured performance is merely an *estimate* of the true AUROC and is subject to *correctness function* errors, which can interact with and distort the final outcome. Specifically, two scenarios may arise:

i) Uncorrelated errors. If errors in $\hat{h}$ are independent of $\hat{g}$, $\widehat{\text{AUROC}}$ is a noisy but *unbiased* estimator of the real AUROC. While scores, in the worst case, regress toward the 0.5 random baseline, no *UQ method* is systematically favored or penalized.

ii) Mutually biased errors. If errors in $\hat{h}$ correlate with $\hat{g}$, the estimated performance of $\hat{g}$ will be *systematically biased*. Depending on the direction of correlation, some *UQ methods* may appear more or less effective than they truly are—leading to inflated or deflated evaluations and ultimately compromising the validity of performance comparisons. These two scenarios are formally characterized and analyzed in App. C, which provides a theoretical foundation for understanding how *correctness function* errors propagate into *UQ performance metrics*.

In practice, in §3 we find that both *UQ methods* and *correctness functions* can exhibit biases over the answer length L, falling into scenario (ii). Understanding the impact of these biases is crucial to ensuring fair and reliable comparisons in UQ.

3 Experiments

In this section, we evaluate several *UQ methods* following evaluation protocols in line with previous related works (Lin et al., 2024; Fadeeva et al., 2023; Farquhar et al., 2024) while varying just the *correctness function*. We consider generative QA tasks, as they are standard in UQ literature and their single-answer format simplifies correctness evaluation relative to more open-ended tasks like summarization.

Experimental setup. We evaluate the performance of 8 *UQ methods* across 4 datasets, 4 models, and 7 *correctness functions*. For details, see: App. A on *UQ methods*; App. B on *correctness functions*; App. D on models, datasets and prompts.

3.1 Impact of the *correctness function* on UQ

Fig. 1 illustrates the estimated performance of different *UQ methods* when varying only the *correctness function*. For each method, we report average $\widehat{\text{AUROC}}$ across datasets and models, focusing on the most commonly used *correctness functions* from Table 1. Fig. 1 reveals that changing the *correctness function* affects not only the estimated AUROC value but also the ranking of *UQ methods*. This, however, raises the question of which metric to trust and what is causing the disagreement.

3.2 Evaluating *correctness functions* for UQ

The previous section shows how *correctness functions* choices impact benchmarking conclusions,

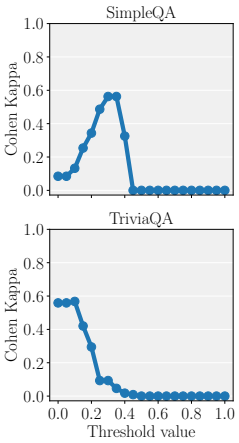


Figure 3: Cohen Kappa score w.r.t. human annotators for Rouge-L (F1) against thresholds.

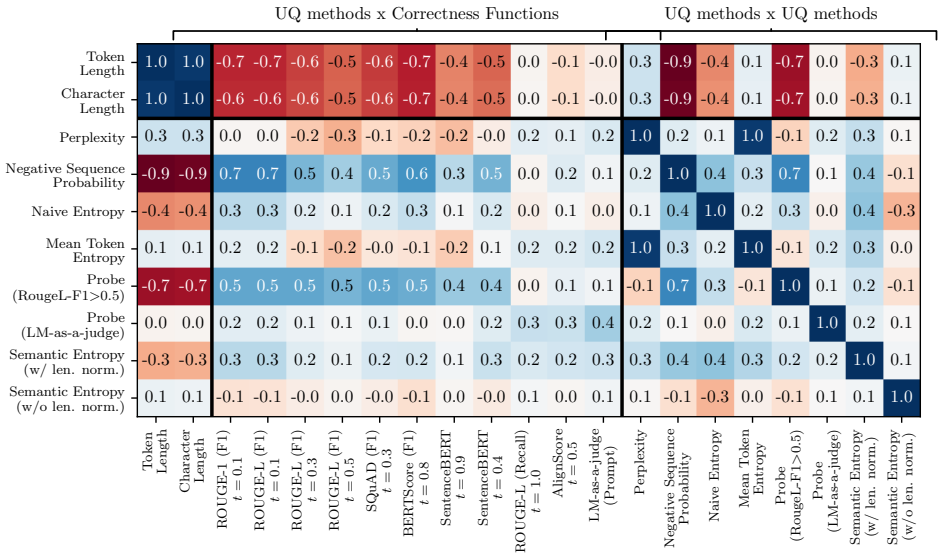


Figure 4: Spearman's rank correlation coefficients.

but it is unclear which function yields reliable UQ results. To investigate, we evaluate several *correctness functions* against human annotations (four annotators per sample for 450 samples, see App. E). The Cohen's Kappa (Cohen, 1960; Artstein and Poesio, 2008) values in Fig. 2[1] show that LM-as-a-judge approaches (prompt-based and AlignScore) align best with human labelers, followed by ROUGE-L (Recall) with $t = 1$.

Revisiting Fig. 1, we see that these three *correctness functions* show more stable orderings, with some variability in AUROC magnitudes—consistent with expectations for small errors that are mostly uncorrelated with *UQ methods*, as per case (i) in §2.4. Conversely, most previously-used lexical- and embedding-based *correctness functions* poorly reflect human judgment.

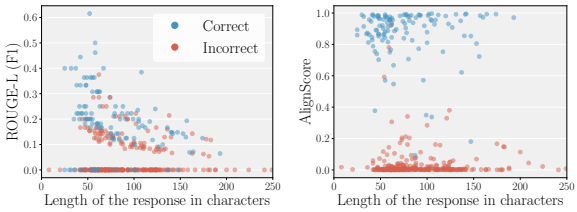
Impact of threshold choice. A major source of error in lexical- and embedding-based *correctness functions* stems from the thresholding strategy used to binarize scores for AUROC computation. As shown in Table 1, prior work often applies standard thresholds or experiments with a small set of options. However, Figs. 2 and 3 illustrate that metrics like ROUGE-L (F1) and SentenceBERT are highly sensitive to threshold choices, as assessed by the resulting agreement with humans. Poor thresholding can lead to degenerate outcomes—e.g., assigning nearly identical labels to all predictions—which drastically reduces alignment with human annotators. The issue is further exacerbated by the fact that optimal thresholds vary across tasks (Figs. 3 and 8) and are heavily influenced by response verbosity (Fig. 5), making it challenging to select a single effective threshold. In contrast, metrics such as ROUGE-L Recall, AlignScore, and LM-as-a-judge exhibit considerably less sensitivity to threshold selection, as shown in Fig. 7 of the Appendix.

3.3 Mutual bias in *correctness functions* and *UQ methods*

We concluded that LM-as-a-judge approaches achieve higher agreement with humans than other *correctness functions*. However, this alone does not explain the shift in *UQ method* rankings observed in §3.1. If errors were random, no systematic effect would emerge, falling into case (i) of our error analysis (§2.4). However, this is not the case: the relative performance of negative sequence probability, perplexity, and probes varies dramatically. This is indicative of a spurious correlation between *UQ methods* and errors in the *correctness functions*.

Many *UQ methods* are biased by length. Many *UQ methods* explicitly or implicitly depend on the length of a response. In particular, negative sequence probability assigns higher uncertainty to longer responses, as each term in Eq. 1 is < 1. Other *UQ methods* that incorporate Eq. 1 in their computation (§2.1), such as Naïve Entropy and Semantic Entropy, may also be impacted. To investigate this relationship, we compute Spearman correlation between the scores from various *UQ meth-*

[1]Some approaches show no agreement at all due to poor thresholding choices.



(a) ROUGE-L (F1) (b) AlignScore

Figure 5: *Correctness function* vs response length. The color indicates human correctness judgments. Results for other *correctness functions* in the Appendix, Fig. 7.

ods and the length of generated answers (measured in tokens and characters). In Fig. 4, we see that multiple estimators exhibit significant positive or negative correlations with length.

Many *correctness functions* are biased by length Many *correctness functions* are also known to exhibit length bias when assessing summaries (Guo and Vosoughi, 2023). We demonstrate that this issue also affects QA. In Fig. 5, we analyze the relationship between response length and *correctness function* output, showing correctness values for responses where all annotators agree on the label. The ROUGE-L (F1) score is highly dependent on response length, favoring shorter sentences and making threshold selection challenging. In contrast, AlignScore is length-independent and clearly separates correct and incorrect samples. App. F presents similar findings for other *correctness functions*.

Spurious interaction. Mutual correlation between *UQ methods* and *correctness functions* on answer length can systematically inflate or deflate *UQ performance metrics* (App. C). This effect is evident in Fig. 1, where length-based baselines (token and character length—blue bars) perform competitively on lexical- and embedding-based *correctness functions* but rank last under LM-as-a-judge metrics. This may also explain discrepancies in prior works, such as the inflated ranking of negative sequence probability in Fadeeva et al. (2023). We recommend using LM-as-a-judge where possible, as its lower error is less likely to impact *UQ performance metrics*. While ROUGE-L (Recall) is inherently independent of the generated answer's length, it offers lower correlation with human judgment, leading to noisier AUROC estimates and offering a higher likelihood of additional confounding variables. For example, it is vulnerable to exploitation by models that produce multiple off-target answers alongside the correct one.

4 Beyond Length Bias

In this paper, we argue that **any biases**—not just length—present **simultaneously** in both the *UQ method* and the *correctness function* induce a spurious correlation that systematically biases the *UQ performance metric* (AUROC). Crucially, this effect does not merely introduce random noise (i.e., increased variance) to the AUROC estimate; rather, it leads to consistent bias, producing misleading results that can artificially favor certain *UQ methods* over others. We support this claim through both analytical derivation (App. C) and empirical evidence (Fig. 1). Importantly, this is a general result. While we use length bias as a running example—because it is is the most severe and already impacting benchmarks—the underlying issue extends to **any confounding variable that correlates with both *UQ methodscorrectness functions***. Identifying such confounders, which are less obvious than length, is inherently difficult, which makes their presence particularly dangerous for evaluation protocols. For instance, in settings that combine LM-as-a-judge approaches with verbalized uncertainty methods (e.g., Huang et al. (2024); Band et al. (2024); Yang et al. (2024)), one might hypothesize less obvious and harder-to-detect sources of confounding like vocabulary used or writing style of the response (Feuer et al., 2025). Our goal is not to enumerate all possible biases but to establish the existence of a broader class of systematic evaluation failures—of which length bias is a concrete and empirically validated case. Identifying and mitigating other such biases remains an important direction for future work.

5 Conclusion

We prove that *UQ performance metrics* are systematically biased when the *UQ method* and *correctness function* share a confounder. Empirically, we identify response length as a concrete instance of such mutual bias which is affecting existing benchmarks and undermines their reliability.

Our results highlight that lexical- and embedding-based *correctness functions*, commonly used in prior work, frequently introduce these distortions. In contrast, LM-as-a-judge approaches exhibit greater robustness and stronger alignment with human judgments, making them a more reliable choice for UQ evaluation. That said, we recommend validating any LM-as-a-judge setup against human annotations before applying it to new tasks or datasets (Bavaresco et al., 2024).

Limitations

In this work, we critically examine the role of the *correctness function* in the evaluation of *UQ methods* using *UQ performance metrics*. While our analysis sheds light on biases introduced by *correctness functions*, certain limitations remain.

Our analysis is focused on the context of QA, as it is a standard task in UQ literature and provides well-defined single-answer questions, making the definition of a *correctness function* easier compared to open-ended tasks like machine translation and summarization where even objective human judgment of correctness is difficult. However, previous work suggests that the length bias of errors in *correctness functions* is not unique to the QA setting (Guo and Vosoughi, 2023), suggesting that *UQ performance metrics* will face similar issues in such tasks.

Our recommendation is to use LM-as-a-judge as a potential *correctness function*. While using another LM to judge correctness has demonstrated advantages (Zheng et al., 2024), it also comes with known limitations (Wang et al., 2024; Chen et al., 2024b). The reliability of the correctness assessment may vary depending on the choice of the judging LM and the prompt formulation. More concerningly, if the same LM is used as both the *correctness function* and as part of the *UQ method*, we are likely to have correlations between the LM-as-a-judge's errors and the *UQ method*, which could inflate the *UQ method*'s performance—although this is mitigated by the relatively low frequency of such errors. Additionally, while our analysis on QA is based on widely used QA datasets, we do not know whether the same LM judge and prompts would generalize effectively to other tasks and datasets. Ideally, an LM judge should be rigorously evaluated against human annotators before being employed in new tasks and datasets (Bavaresco et al., 2024). Furthermore, LM-as-a-judge introduces significant computational overhead compared to traditional *correctness functions*, making it less practical for resource-constrained applications.

Finally, our study identifies response length as a confounding factor in UQ benchmarking, but other latent variables may also influence *UQ methods* and *correctness functions* in subtle ways, as discussed in App. C. A deeper understanding of these biases is crucial for refining UQ evaluation protocols and ensuring more reliable assessments of model uncertainty.

Acknowledgements

We would like to thank Xavier Suau, Miguel Sarabia, Pau Rodríguez, and Eugene Ndiaye for their feedback on an earlier draft of this manuscript.

References

Lukas Aichberger, Kajetan Schweighofer, Mykyta Ielanskyi, and Sepp Hochreiter. 2025. Improving uncertainty estimation through semantically diverse language generation. In *International Conference on Learning Representations*.

Ron Artstein and Massimo Poesio. 2008. Survey article: Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Joris Baan, Nico Daheim, Evgenia Ilia, Dennis Ulmer, Haau-Sing Li, Raquel Fernández, Barbara Plank, Rico Sennrich, Chrysoula Zerva, and Wilker Aziz. 2023. Uncertainty in natural language generation: From theory to applications. *arXiv preprint arXiv:2307.15703*.

Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. 1983. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):179–190.

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. 2024. Linguistic calibration of long-form generations. In *International Conference on Machine Learning*.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, André F. T. Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2024. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. *Preprint*, arXiv:2406.18403.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating question answering evaluation. In *2nd Workshop on Machine Reading for Question Answering*, pages 119–124.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024a. INSIDE: LLMs' internal states retain the power of hallucination detection. In *International Conference on Learning Representations*.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024b. Humans or LLMs as the judge? a study on judgement bias. In *Empirical Methods in Natural Language Processing*.

Arman Cohan and Nazli Goharian. 2016. Revisiting summarization evaluation for scientific articles. In *International Conference on Language Resources and Evaluation*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. 2023. LM-polygraph: Uncertainty estimation for language models. In *Empirical Methods in Natural Language Processing: System Demonstrations*.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Benjamin Feuer, Micah Goldblum, Teresa Datta, Sanjana Nambiar, Raz Besaleli, Samuel Dooley, Max Cembalest, and John P. Dickerson. 2025. Style outweighs substance: Failure modes of LLM judges in alignment benchmarking. *Preprint*, arXiv:2409.15268.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Nuno M Guerreiro, Duarte M Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André FT Martins. 2023. Hallucinations in large multilingual translation models. *Transactions of the Association for Computational Linguistics*, 11:1500–1517.

Xiaobo Guo and Soroush Vosoughi. 2023. Length does matter: Summary length can bias summarization metrics. In *Empirical Methods in Natural Language Processing*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Yukun Huang, Yixin Liu, Raghuveer Thirukovalluru, Arman Cohan, and Bhuwan Dhingra. 2024. Calibrating long-form generations from large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.

Mykyta Ielanskyi, Kajetan Schweighofer, Lukas Aichberger, and Sepp Hochreiter. 2025. Addressing pitfalls in the evaluation of uncertainty estimation methods for natural language generation. *ICLR Workshop: Quantify Uncertainty and Hallucination in Foundation Models: The Next Frontier in Reliable AI*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Association for Computational Linguistics (Volume 1: Long Papers)*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. 2024. Large language models must be taught to know what they don't know. In *Neural Information Processing Systems*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *International Conference on Learning Representations*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Association for Computational Linguistics*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with confidence: Uncertainty quantification for black-box large language models. *Transations on Machine Learning Research*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data only. *Advances in Neural Information Processing Systems*.

Xin Qiu and Risto Miikkulainen. 2024. Semantic density: Uncertainty quantification for large language models through confidence measurement in semantic space. In *Neural Information Processing Systems*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Akim Tsvigun, Daniil Vasilev, Rui Xing, Abdelrahman Boda Sadallah, Kirill Grishchenkov, Sergey Petrakov, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, Maxim Panov, and Artem Shelmanov. 2025. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *Preprint*, arXiv:2406.15627.

Leandro Von Werra, Lewis Tunstall, Abhishek Thakur, Sasha Luccioni, Tristan Thrush, Aleksandra Piktus, Felix Marty, Nazneen Rajani, Victor Mustar, and Helen Ngo. 2022. Evaluate & evaluation on the hub: Better best practices for data and model measurements. In *Empirical Methods in Natural Language Processing: System Demonstrations*.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Association for Computational Linguistics (Volume 1: Long Papers)*.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *Preprint*, arXiv:2411.04368.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *European Chapter of the Association for Computational Linguistics*.

Ruihan Yang, Caiqi Zhang, Zhisong Zhang, Xinting Huang, Sen Yang, Nigel Collier, Dong Yu, and Deqing Yang. 2024. Logu: Long-form generation with uncertainty expressions. *arXiv preprint arXiv:2410.14309*.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Association for Computational Linguistics (Volume 1: Long Papers)*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Neural Information Processing Systems*.

A Details of *UQ methods*

There are several methods for generating uncertainty estimates that help assess the in-correctness of LM outputs. These approaches can be broadly classified into three main categories: **1) Single-sample methods**: methods that require a single forward pass from the model and that generally use directly the logits and probability distributions over the vocabulary space provided as output from the model; **2) Multiple-sample methods**: methods that, given a prompt x, sample multiple possible outputs for the same prompt and compute an uncertainty score based on these outputs; **3) Learned methods**: usually probes or small networks directly trained to predict the accuracy of the model given the prompt and the answer.

We denote with x the sequence of tokens corresponding to the prompt. This usually includes the instruction prompt (e.g., "Answer the following question") together with the question and additional context. The L generated tokens are indicated as $\hat{y}_i$. Additionally, a superscript $\hat{y}^{(s)}$ is used for multiple-sample methods to indicate the s-th sample (out of S_{UQ} samples) sampled for a given prompt. $\hat{p}(\cdot)$ denotes the probability assigned by the model.

Single-sample methods. Single-sample methods estimate the uncertainty score using the logits that the models output. These logits are usually computed on the greedy decoded output or on a low-temperature sample decoded from the model given the prompt x.

Negative Sequence Probability. Sequence probability computes the cumulative probability of the sequence. This can be used as an uncertainty score by flipping the sign and considering $-\hat{p}(\hat{y}|x)$. When evaluated on the greedy decoding samples, this method is sometimes referred to in the literature as *Maximum Sequence Probability* (MSP) (Fadeeva et al., 2023; Vashurin et al., 2025). Aichberger et al. (2025) investigate the difference between the performance of MSP estimated using greedy decoding and estimated using the Beam Search decoding algorithm, which yields sequences with higher likelihood.

$$\hat{p}(\hat{y}|x) = \prod_{i=1}^{L} \hat{p}(\hat{y}_i|\hat{y}_{<i}, x). \tag{4}$$

Perplexity. Perplexity computes the uncertainty score via the exponential of the mean token likelihood (Bahl et al., 1983). Compared to sequence probability, perplexity normalizes the underlying probability by the number of the generated tokens,

$$\exp\left(-\frac{1}{L}\sum_{i=1}^{L} \log \hat{p}(\hat{y}_i|\hat{y}_{<i}, x)\right). \tag{5}$$

Mean Token Entropy. Mean token entropy (Fomicheva et al., 2020; Malinin and Gales, 2020) computes the mean of the per-token entropies over the vocabulary distribution,

$$\mathcal{H}_T(\hat{y}, x) = \frac{1}{L}\sum_{i=1}^{L} \mathcal{H}\left[\hat{p}(\hat{y}_i|\hat{y}_{<i}, x)\right]. \tag{6}$$

Multiple-Sample methods. Multiple-sample methods compute an uncertainty score by sampling S_{UQ} times for a single prompt. Since it is accessing (more of) the full probability distribution, this class of methods should provide better uncertainty scores than single-sample methods, albeit at the expense of an increased computational cost at inference time. The exact number of samples S_{UQ} is a hyperparameter that usually depends on the specific *UQ method*.

Naive Entropy. Naive Entropy computes the entropy over the different generated samples. The sequence probability of each generation is computed using the chain rule of probability, like in the Sequence Probability method,

$$-\sum_{s=0}^{S_{\text{UQ}}} \hat{p}(\hat{y}^{(s)}|x) \log \hat{p}(\hat{y}^{(s)}|x). \tag{7}$$

Semantic Entropy. Semantic entropy computes the entropy over the different semantic clusters C of the generated samples (Farquhar et al., 2024). Semantic clusters are generated using a Natural Language Inference (NLI) model, which evaluates bidirectional entailment between pairs of answers in S_{UQ}. This process group answers with equivalent meanings into clusters $c^{(i)}$. Each cluster probability $\hat{p}(c^{(i)})$ is computed by summing the Sequence Probabilities of the unique generations that fall into that cluster (Farquhar et al., 2024). The probability of each generated sequence is computed using Eq. 4, either directly for Semantic Entropy (without length normalization), or normalized by the sequence token-length L for length-normalized Semantic Entropy.

$$SE(x) = -\sum_{i=1}^{C} \hat{p}(c^{(i)}|x) \log \hat{p}(c^{(i)}|x). \tag{8}$$

Learned methods. Learned methods leverage the model's internal activations or its entire architecture to train additional networks or classifiers that predict the correctness of the answer.

Probes are the most common form of learned method. The most prominent variety of probe is *P(IK)*, also known as P(I Know) (Kadavath et al., 2022), which finetunes the entire model to predicts a binary score whether the model can answer the question correctly or not. This is accomplished by attaching a classifier to the embedding of the final token in the last layer. The training set is collected by labeling some generations from the model with the task *correctness function*. In this paper, we follow the implementation of (Farquhar et al., 2024; Kapoor et al., 2024) that does not train the full model but just a logistic regression classifier on top of the representation of the final answer token as in (Chen et al., 2024a). We trained probes using two different *correctness functions*: LM-as-a-judge (using `Qwen/Qwen2.5-72B-Instruct`), and ROUGE-L (Recall) with a 0.5 threshold. When reporting results, we specify the *correctness function* used to label the dataset and train the probe in round brackets—for example, `Probe(LM-as-a-judge)`, where LM-as-a-judge denotes the judging model. Probes are trained until convergence on each training dataset with L-BFGS and a tolerance value of 0.0001 and maximum number of optimization iterations of 10000.

B Details of *Correctness functions*

In this section, we describe in detail the *correctness functions* used in our experiments. Many of these metrics return a continuous score, which is binarized for calculating AUROC; we detail the thresholds t used in this binarization below.

B.1 Lexical-based

Lexical-based metrics assess similarity by measuring lexical overlap between the generated sentence and the ground truth. These metrics are among the most widely used due to their low computational cost and long-standing history in QA evaluation.

It is important to note that these metrics were originally used to evaluate QA in *trained systems*, where the output distribution of generated sequences has been aligned with the expected distribution of ground-truth answers in the dataset. However, in common LM zero-shot evaluation settings, this alignment is no longer guaranteed. Consequently, these metrics may fail to accurately assess correctness, requiring careful consideration when applying them. While techniques like incorporating few-shot examples, as demonstrated by Farquhar et al. (2024), can mitigate this issue to some extent, this does not fully address the fundamental limitations of lexical-based metrics.

ROUGE-L. ROUGE-L measures the longest common subsequence (LCS) between the generated response and the reference answer, allowing for non-contiguous matches (Lin, 2004). In UQ the F1-score (*ROUGE-L (F1)*) of this metric is typically used, balancing precision and recall (Fadeeva et al., 2023; Kuhn et al., 2023; Duan et al., 2024; Chen et al., 2024a; Qiu and Miikkulainen, 2024; Aichberger et al., 2025). *ROUGE-L (Precision)* measures the ratio of the longest common subsequence (LCS) length to the number of unigrams in the generated answer. *ROUGE-L (Recall)* measures the ratio of the LCS length to the number of unigrams in the reference answer. ROUGE-L (F1) is the harmonic mean of ROUGE-L precision and recall. It is important to note that ROUGE-L recall is not affected by the length of the generated answer, whereas precision and F1 metrics are influenced by it. In the experiments of this paper, we consider *ROUGE-L (F1)* and *ROUGE-L (Recall)* variants, with both metrics computed using the Python package `rouge_scorer`. Both ROUGE-L variants return continuous scores; where a binary score is used, we consider thresholds $t \in \{0.1, 0.3, 0.5\}$ for ROUGE-L (F1), and $t = 1.0$ for ROUGE-L (Recall).

ROUGE-1. ROUGE-1 measures the unigrams overlap between the generated response and the ground truth (Lin, 2004). ROUGE-1 captures similarity based on single-tokens overlap. This metric has been widely used in QA evaluations and in UQ benchmarks in Aichberger et al. (2025). As in Aichberger et al. (2025), we use the F1 variant (*ROUGE-1 (F1)*). In the experiments of this paper, the metric has been computed using the Python package `rouge_scorer`. ROUGE-1 (F1) returns continuous scores; where a binary score is used, we use a threshold $t = 0.1$.

SQuAD. This metric has been introduced in Rajpurkar et al. (2016) to measure the performance of systems trained on the homonymous dataset. The metric computes the F1 score based on word overlap between the prediction and ground truth, treating them as unordered bags of tokens, selecting the

highest F1 among multiple references per question, and averaging across all questions. This metric has been used to evaluate correctness for UQ benchmarks in Farquhar et al. (2024). To compute the metric we used the implementation from Von Werra et al. (2022). SQuAD returns continuous scores; where a binary score is used, we use a threshold $t = 0.3$.

B.2 Embedding-based

Embedding-based metrics assess similarity by encoding both the ground truth and generated text using a neural model, typically BERT-based. The goal is to measure semantic similarity rather than surface-level overlap.

BERTScore. BERTScore (Zhang et al., 2020) evaluates generated answers by embedding both the generated text and the ground truth using a BERT pretrained model. It then computes the pairwise cosine similarity between tokens. For each token in the generated text, the highest similarity score with any token in the reference text is selected. Finally, precision, recall, and F1-score are calculated, with the F1-score commonly used in UQ to balance precision and recall. This metric has been used to evaluate correctness for UQ benchmarks in Fadeeva et al. (2023). In the experiments of this paper, the metric has been computed using the Python package `bert_score` as in Fadeeva et al. (2023). BERTScore returns continuous scores; where a binary score is used, we use a threshold $t = 0.8$, which we empirically found to yield the highest agreement with human raters in Fig. 2.

SentenceBERT Similarity. A SentenceBERT model (Reimers and Gurevych, 2019) is used to encode both the generated answer and the ground truth answer. Specifically, following Chen et al. (2024a), we use *nli-roberta-large*[2]. The cosine similarity is then calculated between the ground truth and generated answer embeddings. This metric has been used to evaluate correctness for UQ benchmarks in Chen et al. (2024a). SentenceBERT returns continuous scores; where a binary score is used, we use a threshold $t \in \{0.4, 0.9\}$.

B.3 LM-as-a-judge methods

LM-as-a-judge metrics evaluate correctness by using another LM to judge the accuracy of a generated answer against the reference answer from the dataset. The evaluating LM may be specifically trained for this task or not.

AlignScore. AlignScore is a metric designed to evaluate the factual consistency of generated text with respect to a ground truth answer (Zha et al., 2023). It employs a RoBERTa model (Liu et al., 2019) trained to assess the alignment between two text pieces, determining how well the generated content corresponds to the source information. The training process integrates data from several NLP tasks—natural language inference, question answering, paraphrasing, fact verification, information retrieval, semantic similarity, and summarization—resulting in a model trained specifically to evaluate correctness. This metric has been used to evaluate correctness for UQ benchmarks in Vashurin et al. (2025). AlignScore returns continuous scores; where a binary score is used, we use a threshold $t = 0.5$.

LM-as-a-judge (Prompt). LM-as-a-judge (Zheng et al., 2024) encompasses a set of approaches that rely on a large language model to provide a human-like assessment of generated content by comparing it against a reference answer. Generally, different prompting strategies can be applied to guide the evaluation process. In our experiments, we used the same prompt as Farquhar et al. (2024) with `Qwen/Qwen2.5-72B-Instruct` as the judging model. LM-as-a-judge returns binary scores, so no thresholds are used.

[2]https://huggingface.co/sentence-transformers/nli-roberta-large

C Impact of Correlated and Uncorrelated Errors in the *Correctness function* on AUROC Estimation

In this section, we analyze the impact of correlated or uncorrelated errors in the *correctness function* on AUROC estimation. We note that a similar analysis is performed in Ielanskyi et al. (2025), a concurrent work that explores impact of bias and variance of *correctness functions* on AUROC estimation.

Let $\hat{g}_i \equiv \hat{g}(\hat{y}_i, x_i) \in \mathbb{R}$ be the uncertainty (UQ) score assigned to the answer $\hat{y}_i$ given the question x_i. We use $h_i \equiv h(\hat{y}_i, x_i) \in \{0, 1\}$ to denote the *ground-truth correctness* of $\hat{y}_i$, i.e., $h_i = 1$ if the answer is correct and 0 otherwise. The *estimated correctness* under some *correctness function* $\hat{h}$ is $\hat{h}_i \equiv \hat{h}(\hat{y}_i, x_i, y_i) \in \{0, 1\}$, possibly using a reference answer y_i.

We define:

$$\text{TPR} = P\big(h = 1 \mid \hat{h} = 1\big), \quad \text{FPR} = 1 - \text{TPR},$$

$$\text{TNR} = P\big(h = 0 \mid \hat{h} = 0\big), \quad \text{FNR} = 1 - \text{TNR}.$$

The *true AUROC* of $\hat{g}$, based on ground-truth labels, is

$$\text{AUROC}(\hat{g}) = P\big(\hat{g}_i < \hat{g}_j \mid h_i = 1,\ h_j = 0\big).$$

When correctness is measured by $\hat{h}$, we obtain

$$\widehat{\text{AUROC}}(\hat{g}) = P\big(\hat{g}_i < \hat{g}_j \mid \hat{h}_i = 1,\ \hat{h}_j = 0\big).$$

We additionally assume $P(\hat{g}_i = \hat{g}_j) = 0$ for $i \neq j$, implying $\text{AUROC}(\hat{g}) = 1 - \text{AUROC}(-\hat{g})$.

Expanding $\widehat{\text{AUROC}}$. We rewrite $\widehat{\text{AUROC}}(\hat{g})$ by conditioning on both the true labels (h_i, h_j) and the estimated labels $(\hat{h}_i, \hat{h}_j)$:

$$\begin{aligned}
&\widehat{\text{AUROC}}(\hat{g}) \\
&= \sum_{a,b \in \{0,1\}} P\big(g_i < g_j \mid \hat{h}_i = 1,\ \hat{h}_j = 0,\ h_i = a,\ h_j = b\big) \\
&\quad \times P\big(h_i = a \mid \hat{h}_i = 1\big) \cdot P\big(h_j = b \mid \hat{h}_j = 0\big)
\end{aligned}$$

We discuss below two cases: (*i*) when $\hat{h}$'s errors are *independent* of $\hat{g}$, and (*ii*) when they are *correlated*.

C.1 Case 1: Independent Errors

Analysis $\hat{h}_i \perp\!\!\!\perp \hat{g}_i \mid h_i$. In this setting, we have

$$\begin{aligned}
&\widehat{\text{AUROC}}(\hat{g}) \\
&= \sum_{a,b \in \{0,1\}} P(g_i < g_j | h_i = a, h_j = b) \\
&\quad \cdot P(h_i = a | \hat{h}_i = 1) \cdot P(h_j = b | \hat{h}_j = 0) \\
&= P(g_i < g_j | h_i = 0, h_j = 0) \cdot \text{FPR} \cdot \text{TNR} \\
&\quad + P(g_i < g_j | h_i = 1, h_j = 1) \cdot \text{TPR} \cdot \text{FNR} \\
&\quad + P(g_i < g_j | h_i = 1, h_j = 0) \cdot \text{TPR} \cdot \text{TNR} \\
&\quad + P(g_i < g_j | h_i = 0, h_j = 1) \cdot \text{FPR} \cdot \text{FNR} \\
&= 0.5 \cdot \text{FPR} \cdot \text{TNR} + 0.5 \cdot \text{TPR} \cdot \text{FNR} \\
&\quad + P(g_i < g_j | h_i = 1, h_j = 0) \cdot \text{TPR} \cdot \text{TNR} \\
&\quad + P(g_i < g_j | h_i = 0, h_j = 1) \cdot \text{FPR} \cdot \text{FNR} \\
&= 0.5 \cdot \text{FPR} \cdot \text{TNR} + 0.5 \cdot \text{TPR} \cdot \text{FNR} \\
&\quad + \text{AUROC}(\hat{g}) \cdot \text{TPR} \cdot \text{TNR} \\
&\quad + (1 - \text{AUROC}(\hat{g})) \cdot \text{FPR} \cdot \text{FNR}.
\end{aligned} \tag{9}$$

All terms $\{\text{TPR}, \text{TNR}, \text{FPR}, \text{FNR}\}$ in Eq. 9 are *constant* properties of the *correctness function* $\hat{h}$, and do not depend on the *UQ method* $\hat{g}$. Hence $\widehat{\text{AUROC}}(\hat{g})$ becomes a "noisy" version of the true $\text{AUROC}(\hat{g})$, biased toward 0.5.

Implications. In this uncorrelated setting, the bias introduced by $\hat{h}$ *does not* depend on the UQ metric $\hat{g}$. Consequently, while the estimated AUROC values will be inaccurate, the ranking of UQ methods by $\widehat{\text{AUROC}}(\hat{g})$ will, in expectation, match the ranking by $\text{AUROC}(\hat{g})$, provided $\text{TPR} \cdot \text{TNR} > \text{FPR} \cdot \text{FNR}$. In practice, finite-sample effects can lead to variance, but with appropriate sample sizes, comparisons based on $\widehat{\text{AUROC}}(\hat{g})$ remain valid.

C.2 Case 2: Correlated errors

Analysis $\hat{h}_i \not\perp\!\!\!\perp \hat{g}_i \mid h_i$. In this case,

$$\begin{aligned}
&P(g_i < g_j | h_i = a, h_j = b, \hat{h}_i = 1, \hat{h}_j = 0) \\
&\quad \neq P(g_i < g_j | h_i = a, h_j = b).
\end{aligned}$$

In particular, if the *correctness function*'s errors are negatively correlated with our *UQ method* (i.e., the more confident the *UQ method* is on a task,

the more likely it is to be erroneously marked as correct), then we have

$$P(g_i < g_j | h_i = a, h_j = b, \hat{h}_i = 1, \hat{h}_j = 0) > P(g_i < g_j | h_i = a, h_j = b),$$

for all values of a and b, with the magnitude of the difference increasing with the magnitude of the correlation. This implies that $\widehat{\text{AUROC}}(\hat{g}) > \text{AUROC}(\hat{g})$. Similarly, if the errors are positively correlated with UQ metric, then we have $\widehat{\text{AUROC}}(\hat{g}) < \text{AUROC}(\hat{g})$.

This indicates that we will over-estimate the true AUROC if we have negatively correlated errors, and under-estimate if we have positively correlated errors. This is problematic because it introduces errors in AUROC that *do* depend on the *UQ method* under consideration, leading to potential reordering of metrics.

Sources of Correlation. Since, in general, the *UQ method* does not depend on the output of the *correctness function* or vice versa, any correlation between the *UQ method*, and errors in the *correctness function*, must be due to information in $\hat{y}$ and/or x that a) introduces systemic errors in the *correctness function*, and b) is used by the *UQ method*. In this paper, we look at length as such a confounding variable, but it is not the only possible option. For example, the use of less frequently occurring words in $\hat{y}$ might lead to both an increase in uncertainty scores due to unfamiliar language, and an increase in the probability of erroneously marking an answer as incorrect due to reduced lexical overlap with the reference answer. We leave the exploration of additional confounders as future work.

D Experimental Details

The datasets considered are `TriviaQA` (Joshi et al., 2017), `SQuAD` (Rajpurkar et al., 2016), `NQ-Open` (Lee et al., 2019), and `SimpleQA` (Wei et al., 2024). The models considered are `Falcon-7B` (Penedo et al., 2023), `Qwen2.5-7B` (Qwen et al., 2025), and two versions of `Mistral-7B` (Jiang et al., 2023).

Our evaluation setup closely follows the methodology proposed by Farquhar et al. (2024)[3]. To obtain model responses, we employed the same prompt as in Farquhar et al. (2024) for the long-form setting, instructing the model as follows:

> *Answer the following question in a single brief but complete sentence.*

Responses are sampled using greedy decoding. Similarly, for the LM-as-a-judge evaluation, we adhered to the same prompt of Farquhar et al. (2024) and used the model `Qwen/Qwen2.5-72B-Instruct` as the judging model. For our experiments, we employed the following models from the Hugging Face Hub `mistralai/Mistral-7B-Instruct-v0.1`, `mistralai/Mistral-7B-Instruct-v0.3`, `Qwen/Qwen2.5-7B-Instruct`, and `tiiuae/falcon-7b-instruct`. The datasets used in our evaluation consist primarily of closed-book QA datasets, with the exception of `SQuAD` which is an open-book dataset. Specifically, for `SQuAD` we incorporated the available context as part of the prompt. In semantic clustering-based methods (Semantic Entropy), we employed DeBERTa as our Natural Language Inference (NLI) module as in Farquhar et al. (2024).

E Details on Human Annotation process

We used an internal crowdsourcing platform to gather annotations. The raters were fluent English speakers and were compensated at or above the minimum wage. We randomly sampled 450 data points from `TriviaQA` (Joshi et al., 2017), `NQ-Open` (Lee et al., 2019), and `SimpleQA` (Wei et al., 2024). We then generated answers using `Qwen2.5-7B-Instruct`, with with greedy decoding. We excluded `SQuAD` from the human annotation process to avoid incorporating the additional context into the annotation prompt, thereby streamlining and accelerating the annotation process. We then tasked human annotators to evaluate the correctness, collecting four annotations per data point. Below, we present the annotation guidelines provided to each annotator. Each dataset included in the guidelines two manually labeled examples.

In Fig. 2 and Fig. 6, we present the Cohen's Kappa agreement rates among human annotators. The first figure reports agreement computed across all 450 data points, while the second breaks down the agreement rates for each individual dataset (150 data points each). For clearer visualization, Fig. 5 and Fig. 7 display a uniformly sampled subset of 150 data points from the full set of 450. These points represent correctness values for responses where all annotators agreed on the label.

[3] https://github.com/jlko/semantic_uncertainty/

(a) TriviaQA (b) SimpleQA (c) NQ

Figure 6: Cohen Kappa agreement rates between human annotators and *correctness functions*.

Annotation Guidelines

You will be provided with three pieces of text:

- **Question**: A question previously proposed to a system or user. You are not asked to answer this question; it has already been answered (see Proposed Answer).
- **Reference Answer**: The authoritative or "gold standard" answer provided for the Question.
- **Proposed Answer**: The response given by a user for the Question that needs to be evaluated.

Your task is to determine whether the **Proposed Answer** is **equivalent** to the **Reference Answer** in the context of the **Question**.

Evaluation Criteria

- **Focus on Equivalence**: Assess whether the Proposed Answer conveys the same meaning as the Reference Answer, regardless of additional details or alternative phrasings.
- **Ignore Personal Knowledge**: Do not rely on your own knowledge or conduct external research. Base your judgment solely on the given text.
- **Exact Matching is Not Required**: The Proposed Answer does not need to be identical to the Reference Answer, but it must preserve the core meaning.
- **Context Matters**: Ensure that the Proposed Answer is relevant to the Question and correctly aligns with the Reference Answer's meaning.

Rating Scale

Choose one of the following ratings for each evaluation:

- **Equivalent**: The Proposed Answer conveys the same meaning as the Reference Answer.
- **Not Equivalent**: The Proposed Answer does not convey the same meaning, either due to missing essential information, contradictions, or incorrect interpretation.

Additional Notes

- **More Detail vs. Different Information**: Extra information is acceptable as long as the meaning remains the same. However, if the additional details introduce contradictions, the Proposed Answer should be marked **Not Equivalent**.
- **Paraphrasing is Allowed**: The wording of the Proposed Answer does not need to match exactly, but the core meaning must remain intact.
- **Avoid Assumptions**: Do not infer additional meaning beyond what is explicitly stated.

Examples

Example 1: Equivalent

<Follows one annotated equivalent example from the dataset>

Example 2: Not Equivalent

<Follows one annotated not equivalent example from the dataset>

F Additional Results

We present here supplementary results that were excluded from the main paper.

Fig. 6 presents the Cohen's Kappa agreement rate with human annotators, broken down by dataset. LM-as-a-judge approaches demonstrate stronger alignment with human judgments, whereas lexical-based and embedding-based *correctness functions* are highly sensitive to the selection of an appropriate threshold.

Fig. 7 illustrates the score assigned by *correctness functions* as a function of the generated answer's length. Among the evaluated approaches, LM-as-a-judge methods (AlignScore and LM-as-a-judge `Qwen/Qwen2.5-72B-Instruct`) appear to be the only robust ones that remain invariant to length while effectively distinguishing between correct and incorrect samples without requiring threshold tuning.

Fig. 8 shows how human-agreement rates vary with the threshold used to binarize different *correctness functions* across datasets. Lexical and embedding-based metrics (e.g., ROUGE-L (F1), BERTScore) show high sensitivity to threshold tuning and inconsistent alignment with human judgments. In contrast, AlignScore yields consistently high agreement across thresholds and datasets.

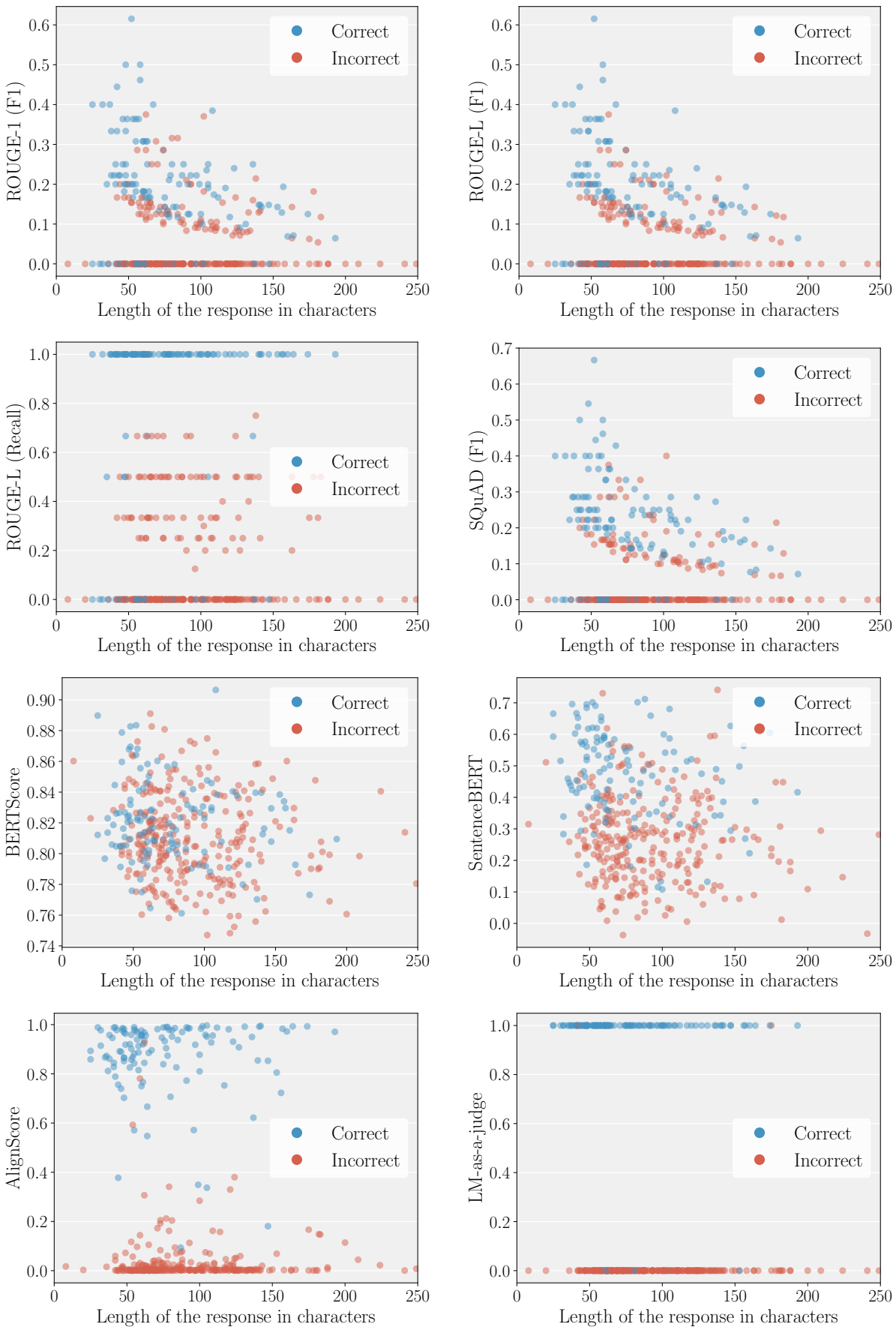


Figure 7: *Correctness function* vs response length. Color indicates human correctness judgments.

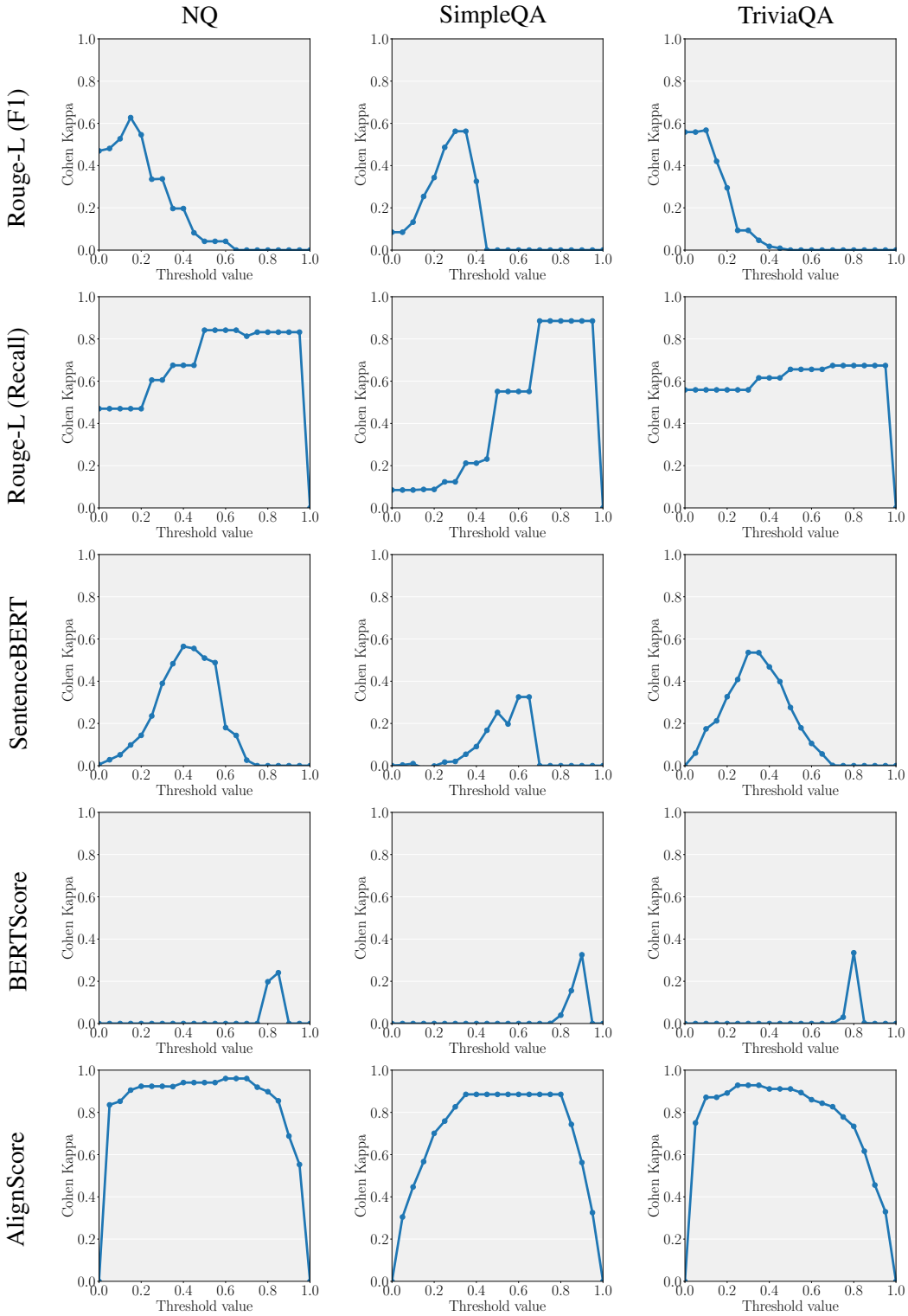


Figure 8: Human-agreement rate as a function of the *correctness function* threshold.

Memorization Inheritance in Sequence-Level Knowledge Distillation for Neural Machine Translation

Verna Dankers[*]
University of Edinburgh
vernadankers@gmail.com

Vikas Raunak[‡]
Microsoft
viraunak@microsoft.com

Abstract

In this work, we explore how instance-level memorization in the *teacher* Neural Machine Translation (NMT) model gets inherited by the *student* model in sequence-level knowledge distillation (SeqKD). We find that despite not directly seeing the original training data, students memorize more than baseline models (models of the same size, trained on the original data)—3.4% for exact matches and 57% for extractive memorization—and show increased hallucination rates. Further, under this SeqKD setting, we also characterize how students behave on specific training data subgroups, such as subgroups with low quality or specific counterfactual memorization (CM) scores, and find that students exhibit greater denoising on low-quality subgroups. Finally, we propose a modification to SeqKD named Adaptive-SeqKD, which intervenes in SeqKD to reduce memorization and hallucinations. Overall, we recommend caution when applying SeqKD: students inherit both their teachers' superior performance *and* their fault modes, thereby requiring active monitoring.

1 Introduction

Memorization of noisy training data creates unexpected failure modes in *neural machine translation* (NMT) models (Raunak and Menezes, 2022), thus presenting a reliability risk when deploying them in the real world. To make NMT models inference-friendly, they are often trained using *sequence-level knowledge distillation* (SeqKD) (e.g., Bapna et al., 2022; Costa-jussà et al., 2022). This is a KD variant in which teachers generate synthetic targets for students (Hinton, 2014; Kim and Rush, 2016). SeqKD yields smaller student models whose performance is competitive with the teacher's performance. Follow-up work has focused primarily on modifying SeqKD objectives to further improve NMT performance (e.g., Wen et al., 2023; Zhang et al., 2023; Wang et al., 2023, 2024). Apart from being used for model compression, SeqKD has proved very beneficial for low-resource and long-tail data (Dabre and Fujita, 2020; Currey et al., 2020; Gumma et al., 2023; Zhou et al., 2024; De Gibert et al., 2024) but its applications extend beyond that as well, e.g., to continual learning (Chuang et al., 2020; Zhao et al., 2022).

☹ Extractive Memorization (ExMem) with respect to the initial parallel corpus increases 57.0%±15.4 in students compared to baselines (i.e. models memorized to emit the target even if we omit the italicized text):

(1) s_C Reprezentacja Trynidadu i Tobago *[w piłce nożnej]*
t_T Trinidad and Tobago national football team
t_S Trinidad and Tobago national football team

☹ Students have 31.0%±25.7 more oscillatory hallucinations (in blue) than baselines:

(2) s_C 1–5 , Stewards are appointed to publish the revelations (...)
t_T 1–5, Diener werden ernannt, um die Offenbarungen zu veröffentlichen (...)
t_S Die Heiligen sind in der Regel in der Regel in der Regel (...)

☹ Students show secondary ExMem (ExMem with respect to the teacher-generated corpus):

(3) s_C Electrical industry in Dominican *[Republic - AmarillasLatinas.net]*
t_T Elektrische Industrie in Dominikanische Republik
t_S Elektrische Industrie in Dominikanische Republik - AmarillasLatinas.net

☺ For low-quality source-target pairs, we observe amplified denoising in students:

(4) s_C La fiche du Pikauba par la Fromagerie Hamel.
t_C Pule » Teuerster Käse der Welt aus Eselsmiclh.
t_T Die Käserei Hamel in Pikauba. (Comet-QE-22=0.47)
t_S Die Geschichte des Pikauba durch die Hamel Käserei. (Comet-QE-22=0.62)
t_B Die Pikauba-Fassung wird von der Käserei Hamel betrieben. (Comet-QE-22=0.47)

Figure 1: An illustration of our findings. Sources (s) are from the corpus (C); translations (t) are from teachers, students and baselines (T, S, B).

[*]Work conducted during an internship at Microsoft.
[‡]Now at Google DeepMind.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 760–774
July 27 - August 1, 2025

Yet, the *understanding* of SeqKD lags behind its *usage*. Prior work in this direction primarily studies why SeqKD is successful, attributing it to mode reduction of the training data (Zhou et al., 2020; Song et al., 2021), or suggesting that SeqKD acts as a regularization technique (Gordon and Duh, 2019). In this work, we better try to understand how model behavior gets transmitted from teacher to student, moving beyond only analyzing average-case performance. In particular, we focus on how the student inherits instance-level memorization. Recent work on image classification suggests that KD inhibits memorization in the student (Lukasik et al., 2024), but also that membership inference attacks on the student are often successful (Jagielski et al., 2024). However, the connection between memorization and SeqKD in NLP is new territory; characterizing this is imperative to mitigate memorization-related failures in students.

Our main contributions are as follows: (1) We provide a quantification of **memorization inheritance** in §2: we identify that even though SeqKD inhibits memorization from teacher to student, the student memorizes more about the initial parallel corpus and hallucinates more than it would have, had it been trained without SeqKD. (2) We perform **subgroup analyses** in §3, for data subsets with specific characteristics, such as examples of different quality levels or with specific counterfactual memorization scores (Feldman, 2020). We identify subgroups for which the student outperforms both the teacher and baseline models. (3) To reduce memorization and mitigate accentuated hallucinations we propose **Adaptive-SeqKD** in §4: a simple intervention in the SeqKD algorithm wherein we adapt the teacher by finetuning it briefly on *intrinsically* obtained high-quality data to reduce memorization and hallucinations in the student.

Figure 1 demonstrates a subset of these findings with examples from our datasets.

2 Memorization inheritance

2.1 Experimental setup

In SeqKD, teacher θ_T is trained on a corpus with source sequences $\mathcal{S}_C$ and targets $\mathcal{T}_C$, and generates translations of $\mathcal{S}_C$ with beam size k. The source sequences and the teacher-generated translations ($\mathcal{S}_C$ and $\mathcal{T}_T$) form the training corpus for student θ_S. We use data from the WMT20 corpus (Barrault et al., 2020), for five language pairs: DE-EN and EN-DE (48M), PL-EN and EN-PL (12M), and FR-DE (14M). Appendix A provides more detail about the WMT corpora, and the validation and test data.

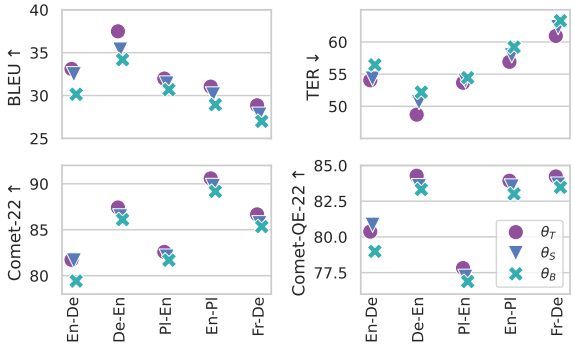


Figure 2: Performance of teacher, student and baseline models for four model quality metrics.

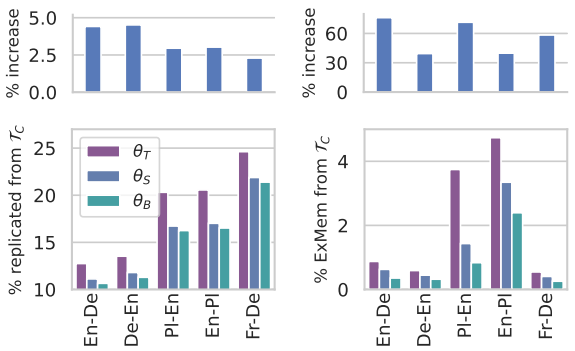
For all languages, a Transformer-large teacher trains for 300k steps on $\mathcal{S}_C$ and $\mathcal{T}_C$, followed by training a Transformer-base student for 100k steps on $\mathcal{S}_C$ and $\mathcal{T}_T$ (k=1).[1] We also train Transformer-base directly on $\mathcal{S}_C$ and $\mathcal{T}_C$, to have a baseline (θ_B) for what the student model would have memorized when exposed directly to WMT20 data. In Appendix C, we furthermore experiment with varying beam size k and the student's model size.

Model quality metrics We first examine the models' quality, to ascertain the SeqKD framework works as intended. We report the following reference-based metrics for translations generated with beam size five: **BLEU** and **chrF**, Translation Error Rate (**TER**), and the **Comet-20** and **Comet-22** metrics that use neural methods for translation quality estimation (Rei et al., 2020, 2022). We supplement this with the reference-free metrics of **Comet-QE-20**, **Comet-QE-22**.

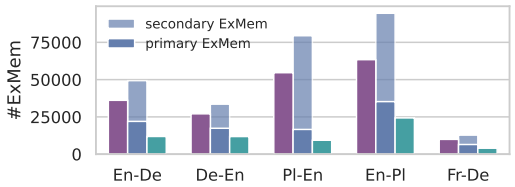
All metrics will be applied to WMT test data, and the reference-free methods are furthermore applied to translations of monolingual data: Common-Crawl data provided by Barrault et al. (2020), and data from the Pulpo poetry corpus (De la Rosa et al., 2023), to examine out-of-domain performance.

Memorization metrics We quantify memorization by comparing greedily translated $\mathcal{S}_C$ to $\mathcal{T}_C$ for all models, and to $\mathcal{T}_T$, for θ_S. We measure the **replication** (exact match) rate, and the **extractive memorization** (ExMem, Raunak and Menezes, 2022) rate. ExMem finds examples for which models memorized to emit the target after seeing at most

[1] Training duration set to equal the setup of Vaswani et al. (2017). We train using `MarianNMT` (Junczys-Dowmunt et al., 2018). For the full training setup see Appendix A and our codebase: `https://github.com/vernadankers/memseqkd`.

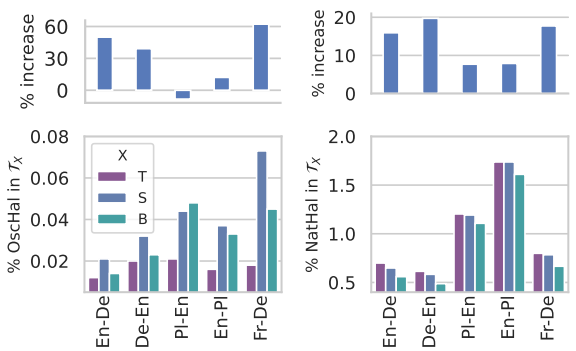


(a) Replication metric (b) ExMem rate



(c) Number of ExMem examples

Figure 3: Memorization metrics for θ_T, θ_S and θ_B and the percentual increase comparing θ_S to θ_B.



(a) OscHal metric (b) NatHal metric

Figure 4: Hallucination metrics for θ_T, θ_S and θ_B and the percentual increase comparing θ_S to θ_B.

75% of the source, e.g., see Example (1). The ExMem rate is the percentage of extractively memorized examples out of the replicated examples.

We also quantify the hallucination rate since hallucinations are often linked to models' memorization capabilities (e.g., Guerreiro et al., 2023; McKenna et al., 2023). We employ binary metrics because they are high-precision and do not require setting a threshold in the absence of ground truth data (Guerreiro et al., 2023; Raunak et al., 2021). We measure the rates of **natural hallucinations** (NatHal) and **oscillatory hallucinations** (OscHal). NatHal is the percentage of source sequences that map to a translation that is repeated in the model's translations at least five times. OscHal is the percentage of translations with bigrams repeated at least 10 times in the target but not the source.[2]

2.2 Results

General model quality Figure 2 and Appendix B provide performance differences for θ_T, θ_S and θ_B. Overall, θ_T outperforms θ_S, and θ_S outperforms θ_B. θ_S and θ_B merely differ in the training targets, which demonstrates that our SeqKD pipeline works as intended. The ordering of models in terms of their quality also holds for CommonCrawl and Pulpo data (see Table 3, Appendix B).

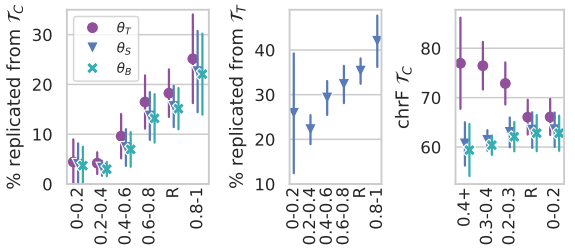
SeqKD facilitates memorization Figure 3 summarizes the memorization results. If we first look at the replication rate, θ_S replicates less from WMT20 than θ_T but **more than** θ_B: students' replication rate with respect to $\mathcal{T}_C$ is 3.4%(±0.9) higher than for θ_B. Students also replicate original material from θ_T: the overall student replication rate for $\mathcal{T}_T$ is 35.3%(±2.7) (see Table 4, Appendix B).

For the ExMem rate with respect to $\mathcal{T}_C$ (Figure 3b), a similar pattern emerges, but with a starker difference between θ_S and θ_B: students extractively memorize less from $\mathcal{T}_C$ compared to θ_T, but **more compared to** θ_B, with a mean increase of 57.0%(±15.4). This is quite surprising; note that by definition, the rate reported here expresses how many of the replicated examples are extractively memorized. Students only observed 18.4% (on average) of $\mathcal{T}_C$ through the SeqKD pipeline (the portion that θ_T replicated) and yet within that smaller pool they still memorized more than θ_B, that was exposed to 100% of the corpus. Not only have students extractively memorized WMT20 examples ('primary ExMem'), they also show ExMem with respect to θ_T ('secondary ExMem', quantified in Table 4, Appendix B). Figure 3c provides the absolute numbers of ExMem examples, distinguishing primary from secondary ExMem that constitute 41% and 59% of all ExMem examples, respectively. Example (3) demonstrates secondary ExMem: the student has memorized to hallucinate "AmarillasLatinas.net" from θ_T's target when merely shown the source's prefix, but θ_T has not.[3]

Why would SeqKD facilitate memorization? We hypothesize that this is due to its denoising function: if noisy data acts as a regularizer during train-

[2]To improve the memorization/hallucination metrics' precision, some training examples are excluded (see Appendix B, along with implementation details and hyperparameters).

[3]ExMem is a more widespread issue affecting commercial translation systems too, see Appendix E.



(a) Quality-based (b) Quality-based (c) CM-based

Figure 5: Illustration of how subgroups (indicated in sub-captions) vary in replication. Error bars indicate standard deviations over language pairs.

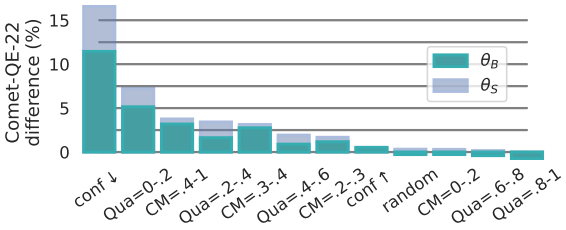


Figure 6: Comet-QE-22 increases compared to the teacher per subgroup, averaged over language pairs.

ing and θ_T partly filters that noise through SeqKD, training θ_S with reduced regularization could lead to increased memorization compared to θ_B. Reduced regularization is traditionally associated with increased memorization in machine learning. In §3, we demonstrate SeqKD's denoising function.

SeqKD amplifies hallucinations It is not just memorization that is amplified through SeqKD; Figure 4 indicates that **hallucinations are also amplified**. For the oscillatory hallucinations, both θ_S and θ_B generate more of these than θ_T, and the student hallucinates more than θ_B, on average (an increase of 31.0%±25.7, with no increase observed for PL-EN). Appendix B presents additional results for monolingual corpora CommonCrawl and Pulpo. Across the board, θ_S hallucinates most there, too.

For the natural hallucinations, θ_S and θ_B hallucinate less than θ_T, but students still hallucinate 13.8%(±5.0) more than θ_B.

3 Subgroup analysis

We now turn our attention to *data subgroups* to describe SeqKD's impact on samples with specific characteristics. We compute replication metrics (**exact match**, **chrF**), neural quality metrics (**Comet(-QE)-22**), and textual diversity (**MSTTR**). Subgroups contain up to 10k examples, with the exception of the random group.

Subgroups We construct 12 subgroups: Firstly, we randomly sample 50k examples from each of the WMT20 corpora. Secondly, we construct five **quality-based subgroups** by bucketing WMT20 examples based on the Comet-QE-22 metric, yielding five subgroups (with ranges <0.2, 0.2-0.4, 0.4-0.6, ≥0.8). Thirdly, we create **counterfactual memorization (CM) subgroups**. Assuming input x and target y, and a model with the parameters θ^{tr} trained on all training data, and θ^{tst} trained on all examples except (x, y), $\texttt{CM}(x, y) = p_{\theta^{\text{tr}}}(y|x) - p_{\theta^{\text{tst}}}(y|x)$ (Feldman, 2020; Feldman and Zhang, 2020). We approximate CM scores for all datapoints for EN-DE, and for 10% of the datapoints for the remaining language pairs, as detailed in Appendix F.1. We create four subgroups of examples (with ranges <0.2, 0.2-0.3, 0.3-0.4, ≥0.4). Finally, we create two **confidence-based subgroups**, by taking θ_T's log-probability averaged over the generated tokens for $\mathcal{T}_T$, selecting the top and bottom 10k examples.

Results We refer the reader to Appendix F for the full results, among which we consider the following patterns to be the most noteworthy:

(1) The random subgroup reflects memorization results: Exact and non-exact match-based metrics for this group (see chrF in Figure 9) follow the overall ranking of θ_T>θ_S>θ_B.

(2) Quality-based subgroups demonstrate (amplified) denoising: The lower the quality of the subgroup, the less models replicate from the corpus (Figure 5a,b). When we look at the Comet-QE-22 quality of the translations the models generate for low-quality subgroups, θ_S and θ_B perform better than θ_T, and θ_S outperforms θ_B (Figure 6). This effect is thus partially attributable to the capacity gap between large and base models, and partially to KD, and confirms prior work (e.g., Zhou et al., 2020) that suggested SeqKD has a denoising function: θ_T denoised WMT20 for θ_S, and now θ_S shows **amplified denoising** on that same data. Example (4) already illustrated this in the introduction.

(3) High counterfactual memorization examples are not replicated: Examples that have high CM for θ_T, do not stand out in terms of replication rates for students or baselines (Figure 5c), and show some amplified denoising (see Figure 6, although substantially less than low-quality subgroups).

(4) Low-confidence examples are not inherited: Finally, for examples for which θ_T generated translations with very low confidence, θ_S has low replication and strong amplified denoising (Figure 6).

Metric	$\theta_{T_{hq}}$	$\theta_{T_{ra}}$	$\theta_{S_{hq}}$	$\theta_{S_{ra}}$
BLEU	$0.0_{\pm 0.5}$	$-1.2_{\pm 0.8}$	$-0.2_{\pm 1.7}$	$-1.2_{\pm 1.6}$
C-QE-22	$+0.2_{\pm 0.3}$	$-0.2_{\pm 0.1}$	$+0.3_{\pm 0.3}$	$-0.1_{\pm 0.2}$
ExMem $\mathcal{T}_C$	$-13.1_{\pm 10.2}$	$-7.5_{\pm 9.3}$	$-23.7_{\pm 24.0}$	$-11.9_{\pm 29.5}$
OscHal	$-55.1_{\pm 11.3}$	$+7.7_{\pm 11.2}$	$-58.9_{\pm 7.5}$	$+9.8_{\pm 15.6}$

Table 1: Percentual performance change following Adaptive-SeqKD, compared to θ_T and θ_S, using high-quality (*hq*) or random (*ra*) data.

4 Adaptive-SeqKD reduces memorization

We previously observed increased memorization and hallucinations in θ_S compared to θ_B, but also amplified denoising on data subsets. We now investigate whether we can reduce students' failures without harming performance otherwise. We apply **Adaptive-SeqKD**, which briefly finetunes the teacher on high-quality data before producing $\mathcal{T}_T$. Instead of costly external metrics like Comet-QE, we use intrinsic metrics for data quality by selecting 500k sequences where θ_T nearly memorized the target (chrF>90), θ_T is confident (normalized translation score > 0.9), and source lengths exceed 5 tokens. We finetuned θ_T per language pair for 200 steps and compared it to finetuning on a random 500k sample meeting the length requirement.

Table 1 shows that teachers finetuned on the high-quality data (and their students) perform similar in terms of BLEU and Comet-QE, but much better in terms of ExMem and OscHal. Finetuning on random data reduced ExMem but not hallucinations. We include additional metrics in Appendix D.

Although we applied Adaptive-SeqKD here by finetuning on a training data subset, the technique could be modified for scenarios in which the training data is unknown. For instance, one could simply take a high-quality subset from a new source and finetune models on it prior to running SeqKD.

5 Related work

Memorization in NLP and NMT Memorization as a general topic has seen increased attention in NLP in recent years. A wide range of work examined how many examples large language models memorize and how to extract those memories (e.g. Carlini et al., 2021, 2023; Nasr et al., 2025), and identified how characteristics of datapoints, models and training techniques relate to memorization (e.g. Mireshghallah et al., 2022; Biderman et al., 2024; Prashanth et al., 2024; Lesci et al., 2024; Li et al., 2024). Alternative lines of related work focused on how memorization affects models internally, for instance, for factual memories (e.g. Geva et al., 2023) or idiomatic expressions (Haviv et al., 2023).

Only a small subset of related work investigated memorization in the NMT context: Raunak et al. (2021) identified that examples with high CM scores tend to elicit hallucinations from models more easily than other examples. In later work, Raunak and Menezes (2022) were the first to discuss the phenomenon of ExMem for NMT systems. Lastly, Dankers et al. (2023) examined the connection between datapoints' features and CM scores in NMT, and emphasized that memorization can sometimes benefit performance.

Knowledge distillation and memorization The connection to memorization and KD had, thus far, only been investigated for the vision domain, where Jagielski et al. (2024) studied membership inference attacks for image classifiers trained through distillation. Although distillation improved the average-case privacy (i.e. reduces memorization), the most vulnerable examples barely benefited from KD. Lukasik et al. (2024) studied CM of image classifiers, concluding that distillation inhibits CM. Some of our findings echo these results, namely that compared to θ_T, memorization is indeed dampened in the student, and that high-CM examples do not stand out in terms of replication.

6 Conclusion

SeqKD is popular for effectively training smaller NMT models, but we show that it also introduces issues like worsened ExMem and hallucinations in θ_S compared to θ_B. At the same time, the subgroup analyses showed that teachers' CM examples are not necessarily replicated by θ_S, and that students exhibit amplified denoising on low-quality examples. This highlights a paradox: through SeqKD, students memorize *more* about the corpus than θ_B, yet also outperform θ_B and θ_T on data where they *did not* memorize. Student improvements thus happen both by mimicking θ_T, and by deviating from θ_T. Future work could suppress memorization during SeqKD, refine Adaptive-SeqKD, and adjust hyperparameters[4] to create more robust SeqKD pipelines. We advise caution with SeqKD: students may inherit not only the teacher's strengths but also its failures, requiring careful monitoring beyond average-case performance.

[4]Increasing k reduces the student's OscHal rate to below that of θ_B, in particular, with Adaptive-SeqKD yielding further improvements (see Appendix C).

7 Limitations

We identify the following three limitations with our work:

- **Limited experimental setup**: inherent to selecting one (although common) experimental setup of distilling a large model into a smaller model, is that the findings need not necessarily transfer to other settings. We experimented with multiple language pairs, and also varied the beam size and student size in Appendix C to comment on this limitation, but recognize that there are other settings that would be interesting to study, such as distilling translation models from LLMs. We opted for the more common SeqKD setup, because of its popularity in the years past; most deployed translation systems are not LLM-based (yet). Besides, SeqKD as a technique is still alive and kicking as demonstrated by, for instance, the recently developed OpusDistillery library[5] (De Gibert et al., 2025).

- **Niche phenomena**: ExMem and hallucinations are phenomena one would only rarely encounter when interacting with NMT systems (Raunak and Menezes, 2022). We consider them worthy of investigation, nonetheless, because they represent extreme system failure that goes far beyond a simple mistranslation. Particularly for high-resource languages, NMT performance is approaching a ceiling, thanks to the sheer volume of translation examples available. Because NMT test performance is so high, it is important that NMT practitioners go beyond standard evaluation and look into domain-specific phenomena, long-tail phenomena (e.g. Raunak et al., 2022), and robustness failures. ExMem and hallucinations represent such failures.

- **Reliability issues not yet solved**: Adaptive-SeqKD substantially reduced the ExMem and hallucination rates, but did not resolve the issue altogether. We conducted limited experimentation in perfecting Adaptive-SeqKD and recognize this could be further expanded upon in the future.

[5] https://github.com/Helsinki-NLP/OpusDistillery

References

Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi Baljekar, Xavier Garcia, Wolfgang Macherey, et al. 2022. Building machine translation systems for the next thousand languages. *arXiv preprint arXiv:2205.03983*.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. 2024. Emergent and predictable memorization in large language models. *Advances in Neural Information Processing Systems*, 36:28072–28090.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Yung-Sung Chuang, Shang-Yu Su, and Yun-Nung Chen. 2020. Lifelong language knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2914–2924.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Anna Currey, Prashant Mathur, and Georgiana Dinu. 2020. Distilling multiple domains for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4500–4511.

Raj Dabre and Atsushi Fujita. 2020. Combining sequence distillation and transfer learning for efficient low-resource neural machine translation models. In *Proceedings of the Fifth Conference on Machine Translation*, pages 492–502.

Verna Dankers, Ivan Titov, and Dieuwke Hupkes. 2023. Memorisation cartography: Mapping out the memorisation-generalisation continuum in neural machine translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8323–8343.

Ona De Gibert, Mikko Aulamo, Yves Scherrer, and Jörg Tiedemann. 2024. Hybrid distillation from RBMT and NMT: Helsinki-NLP's submission to the shared task on translation into low-resource languages of spain. In *Proceedings of the Ninth Conference on Machine Translation*, pages 908–917.

Ona De Gibert, Tommi Nieminen, Yves Scherrer, and Jörg Tiedemann. 2025. OpusDistillery: A configurable end-to-end pipeline for systematic multilingual distillation of open NMT models. In *Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025)*, pages 201–208.

Javier De la Rosa, Álvaro Pérez Pozo, Salvador Ros, and Elena González-Blanco. 2023. Alberti, a multilingual domain specific language model for poetry analysis. *Procesamiento del Lenguaje Natural*, 71:215.

Vitaly Feldman. 2020. Does learning require memorization? A short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959.

Vitaly Feldman and Chiyuan Zhang. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:2881–2891.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235.

Mitchell A Gordon and Kevin Duh. 2019. Explaining sequence-level knowledge distillation as data-augmentation for neural machine translation. *arXiv preprint arXiv:1912.03334*.

Nuno M Guerreiro, Elena Voita, and André FT Martins. 2023. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1059–1075.

Varun Gumma, Raj Dabre, and Pratyush Kumar. 2023. An empirical study of leveraging knowledge distillation for compressing multilingual neural machine translation models. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 103–114.

Adi Haviv, Ido Cohen, Jacob Gidron, Roei Schuster, Yoav Goldberg, and Mor Geva. 2023. Understanding transformer memorization recall through idioms. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 248–264.

Geoffrey Hinton. 2014. Distilling the knowledge in a neural network. In *Deep Learning and Representation Learning Workshop in Conjunction with NIPS*.

Matthew Jagielski, Milad Nasr, Katherine Lee, Christopher A Choquette-Choo, Nicholas Carlini, and Florian Tramer. 2024. Students parrot their teachers: Membership inference on model distillation. *Advances in Neural Information Processing Systems*, 36.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018. Marian: Fast neural machine translation in c++. *Proceedings of ACL 2018, System Demonstrations*, page 116.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Pietro Lesci, Clara Meister, Thomas Hofmann, Andreas Vlachos, and Tiago Pimentel. 2024. Causal estimation of memorisation profiles. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15616–15635.

Bo Li, Qinghua Zhao, and Lijie Wen. 2024. ROME: Memorization insights from text, logits and representation. *arXiv preprint arXiv:2403.00510*.

Michal Lukasik, Vaishnavh Nagarajan, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2024. What do larger image classifiers memorise? *Transactions on Machine Learning Research*.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. In *Conference of the European Chapter of the Association for Computational Linguistics (17: 2023)*, pages 2758–2774.

Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David K Evans, and Taylor Berg-Kirkpatrick. 2022. An empirical analysis of memorization in fine-tuned autoregressive language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1816–1826.

Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Florian Tramèr, and Katherine Lee. 2025. Scalable extraction of training data from aligned, production language

models. In *The Thirteenth International Conference on Learning Representations*.

USVSN Sai Prashanth, Alvin Deng, Kyle O'Brien, Jyothir SV, Mohammad Aflah Khan, Jaydeep Borkar, Christopher A Choquette-Choo, Jacob Ray Fuehne, Stella Biderman, Tracy Ke, et al. 2024. Recite, reconstruct, recollect: Memorization in lms as a multifaceted phenomenon. *arXiv preprint arXiv:2406.17746*.

Vikas Raunak and Arul Menezes. 2022. Finding memo: Extractive memorization in constrained sequence generation tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5153–5162.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183.

Vikas Raunak, Matt Post, and Arul Menezes. 2022. SALTED: A framework for SAlient long-tail translation error detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5163–5179.

Ricardo Rei, José GC De Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André FT Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for mt evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

Yuheng Song, Tianyi Liu, and Weijia Jia. 2021. Data diversification revisited: Why does it work? In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks*, pages 521–533. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jun Wang, Eleftheria Briakou, Hamid Dadkhahi, Rishabh Agarwal, Colin Cherry, and Trevor Cohn. 2024. Don't throw away data: Better sequence knowledge distillation. *arXiv preprint arXiv:2407.10456*.

Shushu Wang, Jing Wu, Kai Fan, Wei Luo, Jun Xiao, and Zhongqiang Huang. 2023. Better simultaneous translation with monotonic knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2334–2349.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10817–10834.

Songming Zhang, Yunlong Liang, Shuaibo Wang, Yufeng Chen, Wenjuan Han, Jian Liu, and Jinan Xu. 2023. Towards understanding and improving knowledge distillation for neural machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8062–8079.

Yang Zhao, Junnan Zhu, Lu Xiang, Jiajun Zhang, Yu Zhou, Feifei Zhai, and Chengqing Zong. 2022. Life-long learning for multilingual neural machine translation with knowledge distillation. *arXiv preprint arXiv:2212.02800*.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in non-autoregressive machine translation. In *International Conference on Learning Representations*.

Yuhang Zhou, Jing Zhu, Paiheng Xu, Xiaoyu Liu, Xiyao Wang, Danai Koutra, Wei Ai, and Furong Huang. 2024. Multi-stage balanced distillation: Addressing long-tail challenges in sequence-level knowledge distillation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3315–3333.

A Data and experimental setup

WMT data We download the parallel corpora from the WMT20 website, using sources listed in Table 2. For EN-DE we use the validation/test data from Raunak and Menezes (2022), which is a combination of WMT test data from recent years. For the other language pairs, we use the WMT20 test data, along with the WMT19 test data for validation during training. We honor the licensing terms by using the WMT data for research purposes only and citing the shared task article.

Source	EN-DE	FR-DE	PL-EN
Europarl	1.8M	1.8M	632k
ParaCrawl	34.4M	7.2M	6.6M
Common Crawl	2.4M	622k	-
News Commentary	362k	284k	-
Wiki Titles	1.4M	942k	1.0M
Tilde Rapid corpus	1.6M	-	278k
WikiMatrix	6.2M	3.4M	3.1M

Table 2: Composition of the 3 parallel corpora.

Additional data We run monolingual model evaluation using data from additional sources:

1. 1M **CommonCrawl** examples, sampled from the first 100M monolingual CommonCrawl datapoints provided by WMT20. To the best of our knowledge, our use is in line with CC's terms of use.
2. Up to 1M **Pulpo** examples, from De la Rosa et al. (2023)'s multilingual Prolific Unannotated Literary Poetry Corpus containing verses and stanzas. Pulpo contains monolingual sequences for all language pairs, apart from monolingual Polish data. We selected the data because it is expected to be out-of-distribution compared to the WMT20 training corpora. Pulpo contains poems that are copyright-free, or distributed under permissive licenses (De la Rosa et al., 2023, p.3).

Training and evaluation Models are trained using the `marian` toolkit (v1.12.16), using the setup of Raunak and Menezes (2022), that mimics the hyperpameters of Vaswani et al. (2017). θ_T trains for 300k steps; θ_S and θ_B train for 100k steps. We use 8 `Tesla V100-SXM2-32GB` GPUs for model training. When evaluating translations using Comet, we use models `wmt20-comet-da`, `wmt22-comet-da`, `wmt20-comet-qe-da` and `wmt22-cometkiwi-da`, using Comet v1.2.0. Visit our git repository for the training, evaluation and visualization code.

B Model quality and memorization

Model quality metrics Table 3 provides model quality metrics for all language pairs. Generally, θ_T>θ_S>θ_B (except for TER, that should be minimized instead of maximized), and where the results differ we italicized the numbers; this mostly happens for the monolingual CommonCrawl data. Yet, across the board, the pattern is clear for both WMT20 and monolingual data. The Pulpo scores are noticeably lower for both Comet-QE metrics. This could indicate that the poetry data is harder to translate, but we cannot reliably make a direct comparison across domains. Most importantly, the system ranking holds for this OOD data, too.

Lang.	θ	WMT20							CommonCrawl		Pulpo	
		BLEU	chrF	TER	Com-20	Com-22	Com-QE-20	Com-QE-22	Com-QE-20	Com-QE-22	Com-QE-20	Com-QE-22
EN-DE	θ_T	33.11	61.46	54.04	*41.77*	81.73	31.52	*80.37*	35.39	*73.11*	11.56	65.83
	θ_S	32.49	61.01	54.31	*42.00*	81.66	31.11	*80.85*	34.93	*74.00*	10.86	65.37
	θ_B	30.15	59.41	56.44	*34.35*	79.42	28.08	*78.99*	34.77	*72.97*	10.58	63.82
DE-EN	θ_T	37.49	64.95	48.70	68.97	87.42	49.40	84.28	38.19	77.01	9.46	64.72
	θ_S	35.36	63.65	50.65	65.87	86.46	46.27	83.55	37.02	76.77	8.37	63.29
	θ_B	34.18	63.11	52.20	64.68	86.11	45.39	83.32	37.02	76.55	8.36	62.54
PL-EN	θ_T	31.98	59.14	53.66	54.20	82.59	34.70	77.82	*34.35*	*74.58*	n/a	n/a
	θ_S	31.41	58.95	53.99	51.83	82.08	32.54	77.21	*34.08*	*74.72*	n/a	n/a
	θ_B	30.69	58.58	54.44	50.55	81.69	32.23	76.88	*34.33*	*74.44*	n/a	n/a
EN-PL	θ_T	31.04	59.86	56.91	92.07	90.59	74.56	83.93	*33.71*	*70.20*	12.24	61.58
	θ_S	30.13	58.98	57.91	89.09	89.77	71.99	83.53	*34.18*	*71.08*	11.46	61.16
	θ_B	28.94	58.27	59.21	86.52	89.17	70.41	83.01	*34.18*	*70.09*	10.94	59.55
FR-DE	θ_T	28.86	60.70	60.94	60.16	86.67	48.24	84.23	33.86	72.24	9.89	58.44
	θ_S	27.77	60.22	62.31	57.08	85.71	46.60	83.66	33.21	72.04	9.42	58.38
	θ_B	26.99	59.69	63.30	55.95	85.34	46.06	83.47	32.90	71.58	9.03	57.30

Table 3: Model performance for the five language pairs, using the experimental setup described in §2.1.

Memorization-related metrics Table 4 provides memorization metrics for all language pairs. For replication, ExMem and NatHal rates, it holds that θ_T>θ_S>θ_B, whereas for oscillatory hallucinations, θ_S typically hallucinates most, although there are exceptions (highlighted in italics).

Lang.	θ	WMT20						CommonCrawl	Pulpo
		Replic. $\mathcal{T}_C$	Replic. $\mathcal{T}_T$	ExMem $\mathcal{T}_C$ (#)	ExMem $\mathcal{T}_T$ (#)	NatHal	OscHal	OscHal	OscHal
EN-DE	θ_T	12.75	n/a	0.875 (36k)	n/a	0.699	0.012	0.020	*0.008*
	θ_S	11.12	32.75	0.627 (22k)	0.397 (49k)	0.648	0.021	0.029	*0.005*
	θ_B	10.65	n/a	0.356 (12k)	n/a	0.559	0.014	0.021	*0.009*
DE-EN	θ_T	13.53	n/a	0.590 (27k)	n/a	0.614	0.020	0.024	*0.062*
	θ_S	11.80	35.17	0.446 (17k)	0.246 (34k)	0.583	0.032	0.037	*0.048*
	θ_B	11.29	n/a	0.320 (12k)	n/a	0.487	0.023	0.026	*0.024*
PL-EN	θ_T	20.31	n/a	3.744 (55k)	n/a	1.203	*0.021*	*0.106*	n/a
	θ_S	16.73	34.65	1.433 (17k)	3.143 (79k)	1.192	*0.044*	*0.135*	n/a
	θ_B	16.25	n/a	0.836 (9k)	n/a	1.107	*0.048*	*0.136*	n/a
EN-PL	θ_T	20.56	n/a	4.736 (63k)	n/a	1.737	0.016	*0.078*	0.017
	θ_S	17.02	33.64	3.347 (35k)	4.126 (94k)	1.737	0.037	*0.100*	0.029
	θ_B	16.52	n/a	2.394 (24k)	n/a	1.610	0.033	*0.126*	0.027
FR-DE	θ_T	24.62	n/a	0.546 (10k)	n/a	0.799	0.018	0.025	0.190
	θ_S	21.88	40.39	0.407 (6k)	0.330 (13k)	0.784	0.073	0.082	0.223
	θ_B	21.39	n/a	0.257 (4k)	n/a	0.666	0.045	0.046	0.081

Table 4: Memorization metrics for the five language pairs, using the experimental setup described in §2.1.

Additional information on ExMem and hallucination metrics To improve the precision of the ExMem and hallucination metrics, some groups of examples are excluded from the computation:

- **ExMem**: in some cases, it is justified that the model emits the target after having processed only 75% of the source, e.g., if the target paraphrases the source. To improve the precision, we thus exclude the following examples when computing ExMem: a) examples with a source shorter than 4 words, b) examples with incorrect source or target languages, c) examples for which the length ratios between source and target are over 1.3, d) examples for which the source equals the target.
- **OscHal**: examples with source sequences of at least 50 white-space tokenized tokens are excluded, because it becomes more likely that the repeated bigrams might be accurate for longer sequences. This only concerns a small portion of the training data, e.g., only 3.4% for EN-DE. We count a sequence as a hallucination if the most frequent bigram appears more than 10 times in the translation, and at least 4 times more often than in the source. We experimentally verified that when reducing that maximum count of 10, the OscHal rate increases, but the model ranking remains the same.
- **NatHal**: for the natural hallucinations, we exclude examples for which the Comet-QE-22 score for the source and the generated translation is above 0.85. This is to exclude cases where natural hallucinations are detected simply because there are source sequences that are each other's paraphrase, for instance if both "Thank you for your visit at our website." and "Thanks for visiting our website." map to "Vielen Dank für Ihren Besuch auf unserer Website.".

C Varying SeqKD hyperparameters

Figure 7 demonstrates how model quality and memorization metrics change when we increase the beam size used to decode $\mathcal{T}_T$ (for both FR-DE and EN-DE), or change the student's model size (for EN-DE). θ_{S_L} and θ_{S_S} have hidden dimensionalities of 1024 and 256, respectively (with corresponding feedforward sizes of 4096 and 1024). When increasing the beam size: i) the quality of the students' translations decrease (for EN-DE) or remain mostly stable (for FR-DE), ii) the replication rates slightly increase, and iii) ExMem decreases, although not consistently: for FR-DE, the students' ExMem rates all exceed the baseline, whereas for DE-EN and $k \in \{2, 5\}$ the students are slightly below θ_B. This is still concerning, considering that students were exposed to 18.4% (on average) of the original corpus, whereas the baseline has seen 100%, so even similar ExMem rates between θ_S and θ_B suggest KD facilitates memorization. The NatHal rate slightly reduces but still far exceeds the baseline. The only real improvement larger beam sizes appear to make are reducing OscHal, both on WMT20 data (OscHal), and on external data

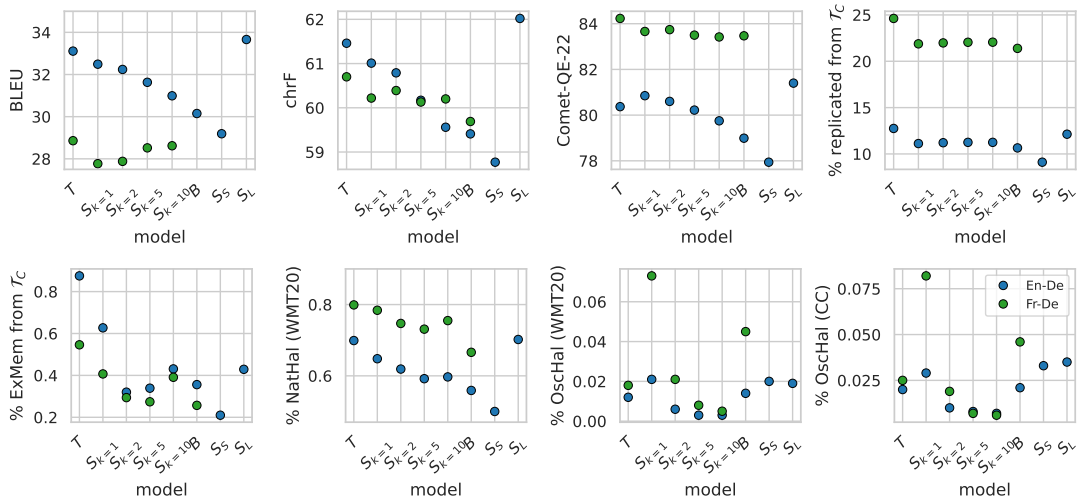


Figure 7: Illustration of how model quality and memorization metrics change as a result of changing the SeqKD beam size for EN-DE.

(OscHal CommonCrawl). With slight performance reduction in EN-DE and no performance reduction in FR-DE this yields a simple KD lesson: even if beam search is more computationally expensive (particularly when applied to the *entire* training corpus): do not use greedy search in KD.

Reducing the student's model size reduces the model's translation quality, and reduces replication, ExMem and NatHal, but still yields similar OscHal rates. The larger model produces better translations than its teacher and has lower ExMem, but also increased hallucinations.

D Adaptive-SeqKD

Figure 8 displays how performance changes per language pair due to Adaptive-SeqKD. Changes to quality metrics like BLEU and Comet(-QE)-22 are relatively minor, but the ExMem and hallucination rates are strongly affected. Finetuning with the high-quality data (*hq* in the graphs) decreases ExMem (for four of five language pairs) and hallucinations (all language pairs); finetuning with random data (*ra* in the graphs) is somewhat effective in reducing ExMem, but mostly fails to effectively reduce the hallucination rates.

In the previous section, we observed that increasing the SeqKD beam size is beneficial for the reduction of the hallucination rate. We thus reran finetuning with high-quality data for $S_{k=5}$ for EN-DE to examine whether finetuning also helps for an increased beam size. Upon doing so, the OscHal rate reduced with 28% for the WMT20 data and 33% for the CommonCrawl data, and the NatHat rate reduced with 10% for the WMT20 data, suggesting the wider applicability of Adaptive-SeqKD.

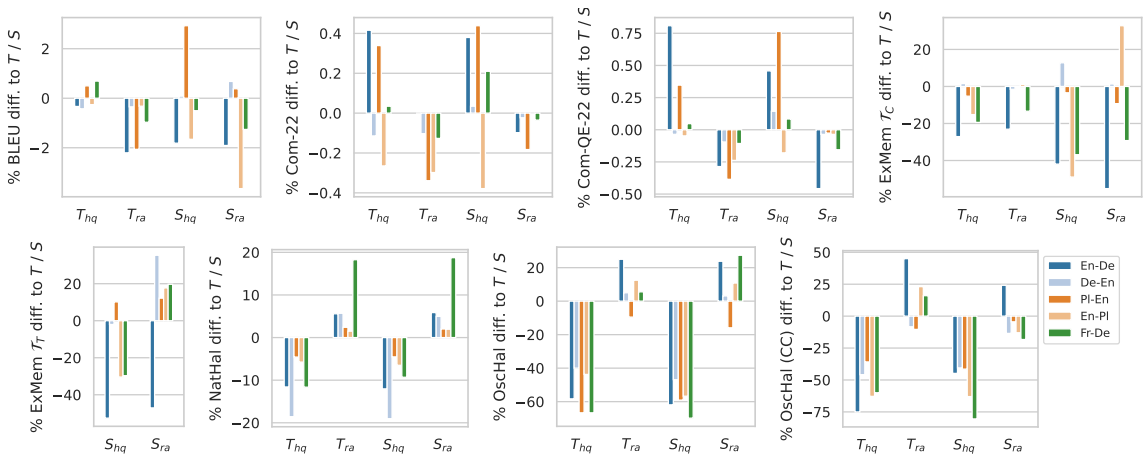


Figure 8: Performance changes observed for the different language pairs when applying Adaptive-SeqKD. Changes are computed as percentages with respect to the original teacher θ_T and θ_S.

E Extractive memorization in commercial systems

In the main paper, we discussed ExMem for the translation systems we trained, but it should be noted that ExMem does not only exist for sytems trained for academic purposes. To assert this, we assessed closed-source translation systems (`Google Translate`, `DeepL` and `Microsoft Bing Translator`),[6] using 70 Europarl examples that led to ExMem for our WMT20 English-German models. ExMem is defined with respect to the training targets, and we do not know what closed-source translation systems are trained on. Still, because of how widespread Europarl is in translation corpora, it may have been a part of the training material for closed-source systems, too. For `DeepL`, the system outputs the target translation without having processed the full source sequence for 25 of those 70 examples. For `Google Translate` and `Microsoft Bing Translator`, that applies to 16 and 20 examples, respectively.

Four examples are provided below. In each source text below, if we omit the italicised part in square brackets, the translations remain the same. Consider the following two examples from `DeepL`:

(5) s I have received seven motions for resolution tabled in accordance with Rule 103(2) *[of the Rules of Procedure]*
t Ich habe sieben Entschließungsanträge erhalten, die gemäß Artikel 103 Absatz 2 der Geschäftsordnung eingereicht wurden

(6) s I have therefore abstained *[from the vote]*
t Ich habe mich daher der Stimme enthalten

Consider the following example from `Microsoft Bing Translator`:

(7) s Mr President, the opposite is *[the case]*
t Herr Präsident, das Gegenteil ist der Fall

Consider the following example from `Google Translate`:

(8) s Mr President, Madam Commissioner, ladies *[and gentlemen]*
t Herr Präsident, Frau Kommissarin, meine Damen und Herren

These are examples copied directly from Europarl, but the underlying robustness issue also pops up when using non-exact variants – e.g. consider the following variant of Example (8): "A cleaner, a teacher and a commissioner said: ladies and". `Google Translate` still translates this including "Meine Damen und Herren" in its translation. It is hard to determine in which other domains ExMem occurs for these models, because we do not have a lot of insight into their training data, but that it is not something that only applies to our models, is for certain.

It is vital that we learn to understand what causes this type of memorization and how to avoid it. We identified knowledge distillation as contributing to exacerbating these types of memorization. We hope our insights can help both scientists and practitioners tackle such problems across different production systems as well, especially since deployed systems tend to be distilled (student) models.

F Subgroup analyses

Before elaborating on the results for the subgroup analysis, we more elaborately explain how we approximated the CM scores.

F.1 Composing the counterfactual memorization subgroups

Assuming input x and target y, and a model with the parameters θ^{tr} trained on all training data, and θ^{tst} trained on all examples except (x, y), CM can be computed as follows (Feldman, 2020; Feldman and Zhang, 2020):

$$\texttt{CM}(x, y) = \underbrace{p_{\theta^{\text{tr}}}(y|x)}_{\texttt{IN}} - \underbrace{p_{\theta^{\text{tst}}}(y|x)}_{\texttt{OUT}}$$

[6] As accessed on the 30th of March, 2025. For `Microsoft Bing Translator`, the 'formal tone' was used for translation.

Leaving out individual datapoints, as this equation suggests, is computationally too expensive given the vast sizes of our WMT20 datasets. We, therefore, approximate CM scores for all datapoints for EN-DE, and for 10% of the datapoints for the remaining language pairs. Following Dankers et al. (2023, p.3), who previously computed CM scores for NMT examples, we replace the probability of the full target with the geometric mean of the target token probabilities. This is more robust to length differences than the full target probability.

For EN-DE, we approximate CM by training 10 teacher models on a randomly sampled 50% of the training corpus, while evaluating it on the remaining 50%, such that for each datapoint, the 'IN' and 'OUT' quantities in this equation are both estimated using five models.

For the remaining language pairs, we use the original θ_T model to estimate the 'IN' quantity, and train another model according to the same training procedure on 90% of the training data, to estimate the 'OUT' quantity for 10% of the remaining training data. This is a rather coarse estimation, but should suffice to determine generic relations between CM and our evaluation metrics of interest.

Using the CM scores, we create six subgroups of interest. The last four were also contained in the main paper, and here we add the first two, that separate examples with a low CM score into two groups:

- IN and OUT performances ≤ 0.2, marked 'L(ow)-L(ow)' in the figures;
- IN and OUT performances ≥ 0.8, marked 'H(igh)-H(igh)' in the figures;
- $\{[0, 0.2), [0.2, 0.3), [0.3, 0.4), [0.4, 1.0]\}$.

F.2 Results

Random subgroup Our baseline subgroup consists of 50k examples randomly sampled from each of the WMT20 corpora. Replication metrics (chrF in Figure 9) follow the overall patterns of θ_T>θ_S>θ_B, consolidating that SeqKD dampens memorization in θ_S compared to θ_T but amplifies it compared to θ_B. The quality and diversity metrics (Comet-QE-22 and MSTTR in Figure 9) emphasize that, compared to the corpus (column $\mathcal{T}_C$), the models generate higher-quality translations, but with lower textual diversity.

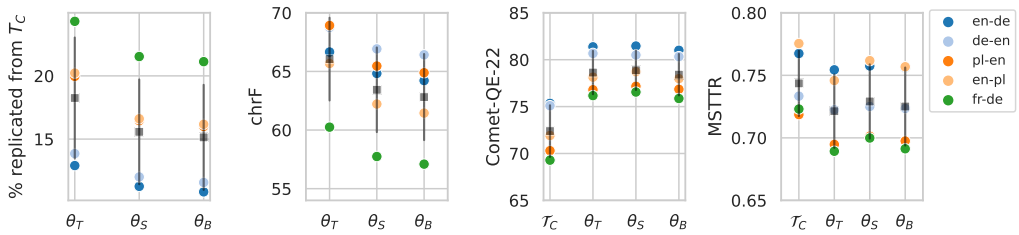


Figure 9: Evaluation metrics applied to the random subgroup, for all five language pairs. The square marker indicates the mean and standard deviation.

Quality-based subgroups How does KD affect examples from the WMT20 corpus with a certain quality? To categorize WMT20 examples based on quality, we applied the Comet-QE-22 metric using the corpus's targets as translations (because the reference is now the translation, we apply Comet's 'reference-free' method). We examine five subgroups: $\{[0, 0.2), [0.2, 0.4), [0.4, 0.6), [0.6, 0.8), [0.8, 1.0]\}$.

In Figure 10, we first notice that for the low-quality subgroups, fewer translations are replicated from $\mathcal{T}_C$ (all models) and from $\mathcal{T}_T$ (for θ_S, see Figure 5b). Secondly, for groups with a quality below 0.6, both θ_S and θ_B generated better translations than θ_T, as per Comet-QE-22 applied to the model-generated translations. Textual diversity metrics here mostly follow the same trend as Comet-QE-22, with θ_S showing the highest diversity.

Figure 13 visualizes the relative differences of students/baselines to the teacher more explicitly, as a percentual increase. For Comet-QE-22, if the student's bar exceeds the baseline's bar, this signifies that the improvement over the teacher is partially attributable to the capacity gap between large and base models, and partially to the SeqKD process. This improvement over the teacher holds for the lower quality groups (0-0.2, 0.2-0.4, 0.4-0.6), but not for the higher quality groups. Example (4) from the introduction already illustrated a sample from the lowest quality subgroup: in the corpus, the target was likely misaligned. θ_T's

translation is slightly better but still wrong, and θ_S's translation is slightly better than θ_B's. Compared to θ_B, θ_S benefits from being presented with the teacher's corpus, which is a **denoised** version of WMT20, since θ_T replicates the least for the lowest-quality (noisiest) examples. θ_S, in turn, replicates less from $\mathcal{T}_T$ for these lowest-quality subgroups, too, which allows for **amplified denoising** compared to θ_T.

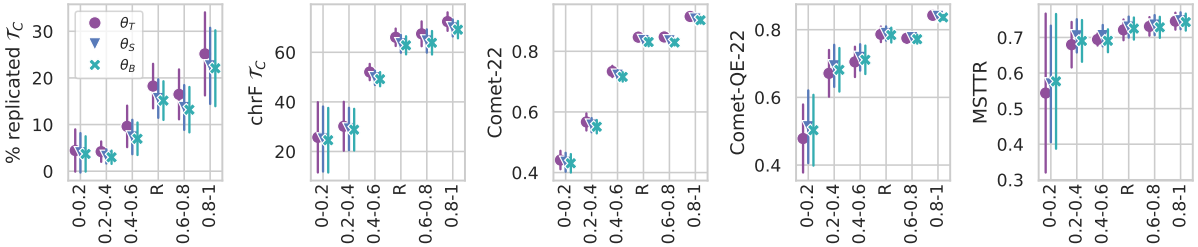


Figure 10: Evaluation metrics applied to the quality subgroups, aggregated over language pairs with error bars indicating standard deviation over language pairs.

Counterfactual memorization subgroups If we now focus on the high CM subgroups (introduced in the previous subsection) in Figure 11, we firstly observe that in terms of memorization metrics, the teacher shows increased replication and chrF. This follows from the definition of CM as the `IN` metric is expected to correlate highly with replication: the higher the target probability, the more likely it is that the teacher can replicate the target when generating translations. We note, however, that the student and baseline replicate examples with high CM less than examples with lower CM. This is likely due to their lower capacity: by definition, CM highlights examples for which a model assigns a low probability to the target when that example is not in the training set, thus requiring more capacity/parameters from the original θ_T to learn that target. The reduced replication also leads to lower Comet-22 scores for these examples, since that metric partially relies on the corpus's target.

When looking at the Comet-QE-22 scores, we observe that the higher CM groups typically also have lower Comet-QE-22 scores, although the quality scores are still above the lowest quality subgroup discussed in the previous paragraph. The baseline and student models outperform θ_T here, likely because they struggle to replicate the somewhat noisy targets as well as the teacher did. The student shows some amplified denoising compared to θ_B, but that does apply to CM groups across the board, and does not seem specific to individual CM subgroups.

All three models show more textual diversity for subgroups with higher CM scores.

Finally, if we inspect the two groups with very low and very high `IN` and `OUT` scores, the models—as expected based on the CM definition—do not replicate the 'low' group, and have very high replication rates for the 'high' group. For the remaining metrics, the 'low' group underscores the findings we previously reported for the low-quality subgroups: the student model shows amplified denoising, as reflected by the Comet-QE-22 metric, and also shows more textual diversity than the other two models. Examples that score badly in terms of both `IN` and `OUT` performance are typically low-quality, misaligned examples (Dankers et al., 2023); if the teacher does not replicate those low-quality targets, but partially denoises by translating them more accurately than the corpus did, the student can further improve upon that.

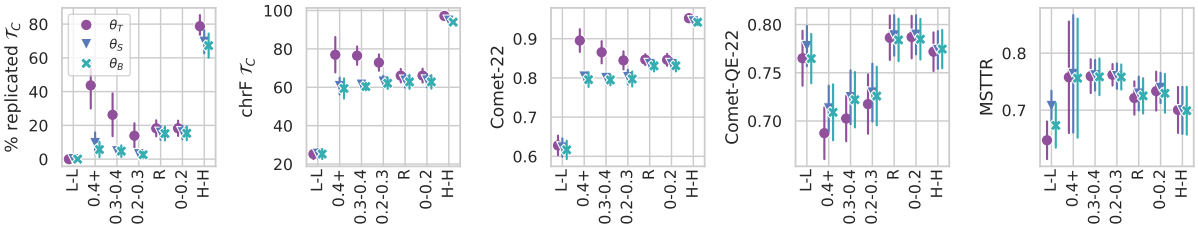


Figure 11: Evaluation metrics applied to the CM subgroups, aggregated over language pairs with error bars indicating standard deviation over language pairs.

Confidence subgroups Finally, we turn to the confidence-based groups, created using the lowest 10k and highest 10k examples based on the mean token-level log-probability of the translations the teacher generated. Figure 12 demonstrates that the low confidence examples stand out for all metrics: none of the models replicate their targets, all three models generate translations with a relatively low Comet-QE-22 score (akin to the lowest-quality subgroups discussed above), and all three models show an exceptionally low MSTTR. If we inspect the examples, they contain many misaligned examples from the corpus for which the teacher also does not generate an adequate translation (e.g., Example (9)), and a lot of cases where the teacher hallucinates in its translation (e.g., Example (10)). The student's improvement here thus does not stem from learning from the superior targets θ_T generated but more likely from deviating from θ_T. Yet, note that even if θ_S does not necessarily copy these hallucinations from its teacher, we did find that the students inspected in the main paper showed amplified hallucination capabilities, so low-quality teacher targets like this could have other unanticipated effects downstream.

(9) s_C Evergreen Terrace - Almost Home 22.
t_C Zum Ernst-Gettke-Haus, Hausnummer 68, siehe unten.
t_T Die alte Burganlage – heute noch 22 Hektar.
t_S Die Evergreen Terrace ist fast die Heimat 22.
t_B In: Evergreen Terrace – Fast Home 22.

(10) s_C Because the race is restricted to Canadian-bred horses, it is not eligible for grading, despite being one of Canada's most prestigious races Northern Dancer Turf Stakes, (...)
t_C Kammerherr, Land- und Obergerichtsrat Magnus Graf von Moltke, Ständedeputierter der Stadt Schleswig, Präsident der konstituierenden Ständeversammlung des Jahres 1836, (...)
t_T Stakes, (...)
t_S Depots of Canada'hintergrund für kanadische Pferde, ist es nicht berechtigt für das Einstufung, obwohl es eines der renommiertesten Rennen Northern Dancer Turf Stakes, (...)
t_B Stakes, (...)

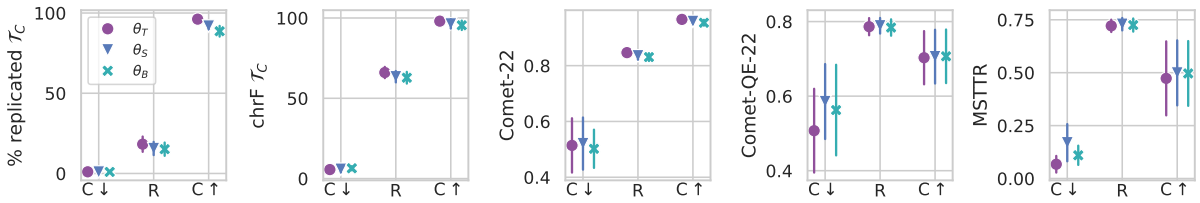


Figure 12: Evaluation metrics applied to the confidence subgroups, aggregated over language pairs with error bars indicating standard deviation over language pairs.

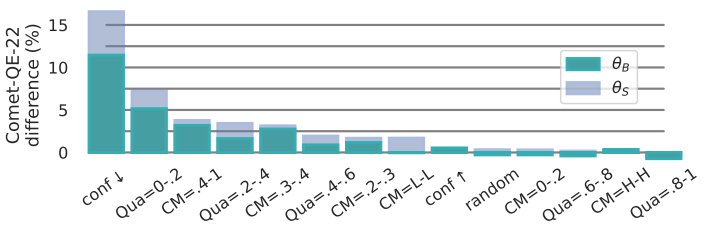


Figure 13: Relative increases comparing students and baselines to the teacher models, for the Comet-22-QE metric.

CoRet: Improved Retriever for Code Editing

Fabio Fehr*
EPFL & Idiap Research Institute
fabio.fehr@epfl.ch

Prabhu Teja Sivaprasad
AWS AI Labs
prbuteja@amazon.de

Luca Franceschi
AWS AI Labs
franuluc@amazon.de

Giovanni Zappella
AWS AI Labs
zappella@amazon.de

Abstract

In this paper, we introduce CoRet, a dense retrieval model designed for code-editing tasks that integrates code semantics, repository structure, and call graph dependencies. The model focuses on retrieving relevant portions of a code repository based on natural language queries such as requests to implement new features or fix bugs. These retrieved code chunks can then be presented to a user or to a second code-editing model or agent. To train CoRet, we propose a loss function explicitly designed for repository-level retrieval. On SWE-bench and Long Code Arena's bug localisation datasets, we show that our model substantially improves retrieval recall by at least 15 percentage points over existing models, and ablate the design choices to show their importance in achieving these results.

1 Introduction

Code editing is an important task that often requires developers to make changes to code repositories based on explicit natural language descriptions such as a GitHub pull request about a bug, a new feature, or about an error in the code. Successful code editing requires correct navigation and retrieval of relevant sections of the repository. This process demands a repository-level understanding of the code functionality (semantics), organisation (repository hierarchy), and relationships between various entities in the codebase (*e.g.* runtime dependencies). Retrieving multiple relevant code chunks is especially difficult in large, real-world repositories (Jimenez et al., 2024). Retrieval is important for both coding agents and humans and forms an important first step in the overall code editing process.

We find that existing pretrained encoder models perform poorly in repository-level retrieval for code-editing. We conjecture the reason for this is threefold: the representations do not align the problem statements to segments of code; the structure from the repository hierarchy is lost; and the runtime dependencies within a repository are not captured. Models like CodeSage (Zhang et al., 2024a) capture docstring-code relationships effectively. Yet, we find that this capacity does not readily transfer to retrieval for code-editing tasks. Similarly, models like CodeBERT (Feng et al., 2020), GraphCodeBERT (Guo et al., 2021), and UniXcoder (Guo et al., 2022) focus on code semantics but miss repository-level structures, limiting their generalisation to this task.

Incorporating additional context has been long known to improve the quality of retrieval (Khattab and Zaharia, 2020; Zhu et al., 2024). The nature of code presents challenges in determining how to integrate context and what type of context to use. For tasks like code completion and summarisation, using chunks from other files within the same repository as context improves performance (Bansal et al., 2021; Ding et al., 2024). Bansal et al. (2023) argue that the call graph dependency structure is required to understand the semantics of code and consider embedding code subroutines together. We follow this direction.

Our work introduces a dense retrieval model for code-editing tasks. We show that optimising directly the likelihood of the model to retrieve correct code sections brings significant advantages over standard contrastive losses. Moreover, we show how to effectively incorporate file hierarchy and call graph context. These contributions lead to improvements of at least 15 percentage points recall over the baseline methods on SWE-bench and Long Code Arena.

2 Problem setup

Representing a code repository in structurally-informed and semantically succinct units is imper-

*Work done during an internship at AWS, Berlin.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 775–789
July 27 - August 1, 2025

ative for retrieval performance (Shrivastava et al., 2023; Zhang et al., 2023b; Liao et al., 2024). An obvious atom of a repository is a file. However, files may be particularly long and may lack semantic coherence. Therefore, we further break down each file into its constituent functions, classes and methods of classes, as illustrated in Figure 1 (bottom). We refer to each such atom as a *chunk*. Further details on the chunking procedure are available in Appendix B.

The objective of code retrieval is to return all parts of the codebase that are pertinent to the current input. Let $\mathcal{C}$ represent a code repo as a set of atoms: $\mathcal{C} = \{c_1, \cdots c_M\}$. We represent with q a natural text input that describes an issue in the repository. Our dataset $\mathcal{D}$ consists of N triplets of issue descriptions, associated codebase, and the ground truth atoms *i.e.* $\mathcal{D} = \{(q_i, \mathcal{C}_i, \mathcal{C}_i^*)\}_{i=1}^N$ where $\mathcal{C}_i^* \subseteq \mathcal{C}_i$. We refer to each triplet as an example, or instance. A retriever is a function $f : (q, c_i) \rightarrow [0, 1]$ for $c_i \in \mathcal{C}$ that assigns a numerical score to each of the atoms in $\mathcal{C}$. A ranking can be induced on the atoms by sorting the scores. We parameterize $f(q, c_i; \theta_q, \theta_c) = \text{sim}(Q(q; \theta_q), C(c_i; \theta_c))$, where Q and C are embedding networks for q and c_i respectively, and 'sim' is the cosine similarity between them. The ideal retriever induces a ranking that puts all the relevant atoms before the irrelevant ones. In practice, we extract the top-k most similar atoms to the query, denoted by $\mathcal{C}_F$, where $k \ll |\mathcal{C}|$ *i.e.* $\mathcal{C}_F = \arg\text{top}\, k_{c \in \mathcal{C}}[f(q, c)]$. We leave the dependence of $\mathcal{C}_F$ on k implicit. For brevity, we represent $Q(q; \theta_q)$ with $\mathbf{q}$ and $C(c_i; \theta_c)$ with $\mathbf{c}_i$. During inference, given a (test) codebase and a query in natural language, the top-k most similar atoms retrieved are from the set $C = \{c_1, \ldots, c_M\}$ where M is the number of atoms (chunks) in this codebase.

3 Proposed method

Here we describe our method to train encoder models for the specific task of code retrieval.

3.1 Training

The goal of training the retrieval model f is to learn an embedding space where the query and the relevant code chunks have higher similarity than irrelevant code chunks. Each instance $i \in [N]$, contains a single query q_i and multiple ground truth code chunks c_j^*. These form positive pairs (q_i, c_j^*). The remaining pairs (q_i, c_k), for $c_k \in \mathcal{C} \setminus \mathcal{C}^*$, are

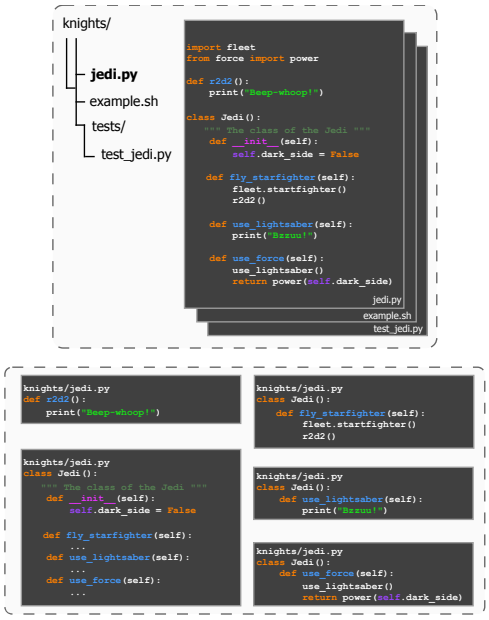


Figure 1: **Top**: Code repository before processing. **Bottom**: Code chunks after filtering and chunking.

negative pairs. We optimise the following loss function:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i}^{N} \frac{1}{|\mathcal{C}_i^*|} \sum_{c^* \in \mathcal{C}_i^*} \log \frac{\exp\left(\frac{\mathbf{q}_i \cdot \mathbf{c}^*}{\tau}\right)}{\Gamma(\mathbf{q}, \mathcal{C}_i)}, \quad (1)$$

where $\theta = (\theta_q, \theta_c)$ are the parameters of the model and $\Gamma(\mathbf{q}, \mathcal{C}) = \sum_{c \in \mathcal{C}} \exp\left(\frac{\mathbf{q} \cdot \mathbf{c}}{\tau}\right)$ is the normalising factor. This is the mean likelihood of retrieving the code chunk per model, and is akin to the standard cross-entropy loss for multi-class classification.

Since the normalising factor involves a summation over all the chunks in a code repository (which can be in the order of 10000 chunks), we implement an approximation by considering only a random subset of instances:

$$\tilde{\Gamma}(q, c^*, \mathcal{C}) = \exp\left(\frac{\mathbf{q}_i \cdot \mathbf{c}^*}{\tau}\right) + \sum_{c \in \mathcal{B}} \exp\left(\frac{\mathbf{q}_i \cdot \mathbf{c}}{\tau}\right) \quad (2)$$

where $\mathcal{B} \subset \mathcal{C}$ is random sample of within-instance negatives and τ is a temperature parameter. Following Zhang et al. (2024a), we set $\tau = 0.05$ throughout. Prior works (see Appendix A) primarily use contrastive losses for feature learning, whereas we apply a standard log-likelihood loss in this setting.

3.2 Call graph context for code-editing

Context has been shown to significantly improve the quality of retrieval (Lewis et al., 2020b; Günther et al., 2024; Borgeaud et al., 2022). A code repository has a natural relationship between its chunks

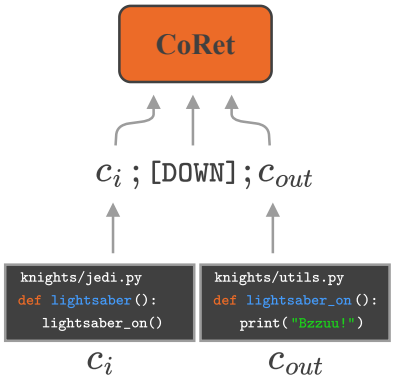


Figure 2: Given a function c_i and its downstream neighbour c_{out}, we concatenate the strings as c_i; [DOWN]; c_{out}, including the special separation token, and fine-tune the model to obtain **CoRet**.

endowed by the call graph (Ahn et al., 2009; Bansal et al., 2023). The neighbours in the call graph are the code entities that are invoked by or invoke the current chunk of interest. We propose to enrich each chunk c_i in $\mathcal{C}$ with its call graph neighbours $\mathcal{N}(c_i)$. To incorporate this information, we modify the chunk embedder C to accept that information as $C(c_i, \mathcal{N}(c_i); \theta_c)$. Several implementations for this C are possible: we retain the same network as without the call graph information, but add the call graph chunks in the input itself *i.e.* $C([c_i; \mathcal{N}(c_i)]; \theta_c)$. We also use only downstream neighbours by introducing a new token [DOWN] that denotes this relationship. For instance, a chunk with one incoming and one outgoing edge is represented as c_i; [DOWN]; c_{out}, where ';' denotes string concatenation. See Figure 2 for an example of how the call graph context is fed to the model. We follow BERT (Devlin et al., 2019) in adding token segment type embeddings *i.e.* trainable embeddings that signify c_i and $\mathcal{N}(c_i)$.

4 Experiments

4.1 Dataset

For training, we consider repository-level code-editing problems which contain a language problem statement q and a code repository $\mathcal{C}$ from SWE-bench (Jimenez et al., 2024). The ground truth pull requests are parsed to obtain the ground truth chunks $\mathcal{C}^*$ which correspond to the edited code chunks. We evaluate our trained models on SWE-bench Verified, and Long Code Arena (LCA) bug localisation (Bogomolov et al., 2024). We provide further dataset statistics in Appendix D.

4.2 Metrics

We measure the performance of our model using standard retrieval metrics: recall@k and mean reciprocal rank (MRR). Recall@k measures how many of the ground truth chunks in $\mathcal{C}^*$ are retrieved when k most similar chunks are retrieved. MRR measures the minimum k needed to retrieve at least one correct code chunk; see Appendix C for formal definitions. Additionally, when multiple chunks have to retrieved, as in the case of LCA, we show the performance metric Perfect-Recall@k which is a binary value for each instance if all correct chunks were retrieved at k. This is better suited to measure improvements as partial retrievals are not useful for subsequent code editing.

$$\text{Perf-Recall@}k = \frac{1}{N}\sum_{i=1}^{N}\begin{cases}1 & \text{if } \mathcal{C}_{Fi} \cap \mathcal{C}_i^* = \mathcal{C}_i^*, \\ 0 & \text{otherwise.}\end{cases} \quad (3)$$

We prioritize recall in our choice of metrics because retrieving all the necessary code chunks is necessary to solve a task, while the impact of retrieving unnecessary chunks (*i.e.* low precision) is not clear.

4.3 Implementation

We implement CoRet using CodeSage Small (S) (Zhang et al., 2024a) as our pre-trained backbone with the following modification: we use mean pooling over all chunk tokens instead of using the standard [CLS] token. This modification resulted in a moderate performance boost in early experiments. We also tie the weights for both C and Q models, meaning $\theta_c = \theta_q$, which we initialise with the publicly available CodeSage S weights. We found that weight tying performs consistently better in our experiments than letting C and Q vary independently during training. Further model and implementation details are in Appendix E and Appendix G.

4.4 Results

Existing models are sub-optimal when used for retrieval in code editing. We compare our model to standard methods such as BM25 (Trotman et al., 2014), several text-code encoder models like CodeBERT (Feng et al., 2020), GraphCodeBERT (Guo et al., 2021), UniXcoder (Guo et al., 2022), and CodeSage (Zhang et al., 2024a). In Figure 3, we present each baseline model retrieval performance

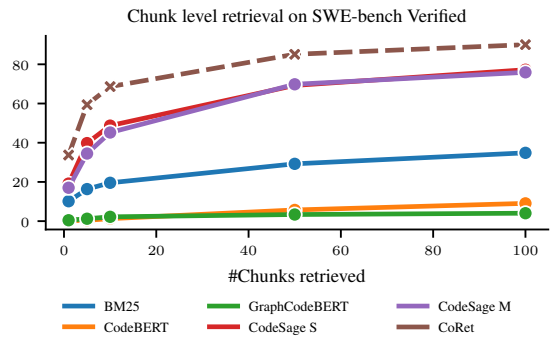


Figure 3: Recall@k for the baseline models and our proposed method on instances of SWE-bench Verified. CodeSage-family models substantially outperform other baselines. CoRet, described in Section 3, consistently outperforms baselines across all k. The dashed line corresponds to the trained CoRet model, whereas the solid lines correspond to the untrained baselines.

	SWE Verified			LCA		
Model	@5	@20	MRR	@5	@20	MRR
CodeSage S	0.34	0.51	0.35	0.26	0.34	0.28
CoRet − CG	0.52	0.69	0.52	**0.32**	0.41	0.45
CoRet − CG + file	**0.54**	0.69	0.52	0.29	0.38	0.44
CoRet	**0.54**	**0.71**	**0.53**	**0.32**	**0.47**	**0.47**

Table 1: Perfect recall at chunk level on code-editing retrieval tasks.

on SWE-bench Verified in solid lines. Among the models considered, CodeSage S and CodeSage M perform the best with the caveat that CodeSage M's performance comes with a much higher computational resource requirements. For this reason, we pick CodeSage S model as our pretrained backbone for further finetuning.

Representation learning focused on retrieval improves performance. In Figure 3, the dashed line reports the performance of CoRet after training. We report the perfect chunk recall in Table 1. It is evident that CoRet improves upon the best baseline CodeSage S significantly. On SWE-bench Verified, recall@5 improves by 52.9% compared to CodeSage S for recall@5 and by about 35% for recall@20.

Call graph context improves multi-chunk retrieval. Next, we ablate on the call graph context to assess the contribution of including this additional information during training and inference. Table 1 reports perfect chunk recall on our two evaluation datasets. CoRet - CG indicates our method without the call graph context described in Section 3.2. We present results of additional baselines in Figure 4

	wo/filepath			w/filepath		
Model	@5	@20	MRR	@5	@20	MRR
BM25	0.15	0.21	0.14	0.16	0.22	0.16
CodeSage S	0.40	**0.58**	0.33	0.40	0.57	0.35
CoRet	**0.42**	**0.58**	**0.42**	**0.53**	**0.70**	**0.53**

Table 2: Chunk-level accuracy on SWE-bench Verified, comparing CoRet without (wo/filepath) and with (w/filepath) file path input. CoRet is fine-tuned with file paths, highlighting performance drop without them. CoRet performance improves with file path context.

in the Appendix and provide a summary here. We further perform an ablation (CoRet - CG + file) where we include as additional context a number of chunks randomly sampled from the same file as the target chunk, and we leave the rest of the pipeline unchanged. We observe a substantial decrease in performances throughout, further validating our design choice.

Choice of negative samples in Equation (2) influences the performance. Traditional methods for representation learning using contrastive losses (Chopra et al., 2005; Chen et al., 2020; Karpukhin et al., 2020) form negative samples by using positives from elements across the batch. We term this *across-instance negatives*. In Equation (2), we form the negatives from entirely within a problem's repository and not across problems, like in Sohn (2016a), and we term this *in-instance negatives*. For our experiments we randomly draw up to a maximum of 1024 negative samples to compute the term $\tilde{\Gamma}$ in our loss (Equation (2)) and consider the influence of the number of negatives in Figure 5 in the Appendix. We find improvements for all k when we include in-instance negatives. Additionally, the impact of number of negatives is also evident; the larger the number of negatives, the better the performance. Increasing the number of negatives from $8 \rightarrow 1024$ improves the recall@20 by almost 10 points. This provides further evidence that Equation (1) with the approximation from Equation (2) mimics the goal of a retriever: it explicitly models which code chunks are relevant and which are not.

File hierarchy is an important feature for retrieval. We prefix the file path to each chunk as a part of the chunking strategy. We find that it is an important feature. In Table 2, we show the retrieval performance drops for CoRet when file paths are removed from the chunk representation for infer-

ence, as the models are originally trained with the file path present. The performance difference is minimal for both BM25 and CodeSage S whereas CoRet learns to rely on the file name through training. Additional evidence from an attention matrix is presented in Appendix I.

5 Conclusions

In this work, we present an explicit study of repository-level code-editing retrieval. We propose a method for training models specifically designed for this task and demonstrate that existing retrieval models are suboptimal in this setting. We identify that this problem differs from the traditional constrastive representation learning problem and propose a loss function that substantially improves the performance of retrieval models compared to using standard contrastive losses. Further, incorporating code context from neighbours in the call graph gives an additional boost in performance. We speculate that further improvement may come with better strategies to select relevant neighbours, *e.g.* by leveraging topological properties of the call graph Tsourakakis et al. (2017); Chiang et al. (2014); Cesa-Bianchi et al. (2012). We hope this work highlights the importance of retrieval for code editing and inspires further research to advance models and techniques in this domain.

Limitations

Our work can be extended in several ways. For instance, our experiments are restricted to Python, and are based on two datasets - SWE-bench and LCA. At the time of the submission they were the only datasets that allowed for repository level code retrieval problems. Recently, a multi-language dataset called SWE-PolyBench (Rashid et al., 2025) has been released and we plan to include it in our future work on this topic.

Furthermore, SWE-bench Verified requires to modify, for the large part, one file in each test case. Thus, file recall can be very high. LCA features edits in multiple files, and thus is a better repository to benchmark. The data preparation and chunking step, while extensible, can be expensive to implement for languages other than Python.

Our investigation has been limited to encoder models for their feature prediction abilities. Several modern LLMs have been modified to output feature embeddings (BehnamGhader et al., 2024; Tao et al., 2024). It is also likely that these LLMs trained on large corpus can provide better baseline performance, but training them requires more resources.

References

Shin-Young Ahn, Sungwon Kang, Jongmoon Baik, and Ho-Jin Choi. 2009. A weighted call graph approach for finding relevant components in source code. *2009 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing*, pages 539–544.

Aakash Bansal, Zachary Eberhart, Zachary Karas, Yu Huang, and Collin McMillan. 2023. Function call graph context encoding for neural source code summarization. *IEEE Transactions on Software Engineering*, 49:4268–4281.

Aakash Bansal, Sakib Haque, and Collin McMillan. 2021. Project-level encoding for neural source code summarization of subroutines. In *2021 IEEE/ACM 29th International Conference on Program Comprehension (ICPC)*, pages 253–264.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*.

Egor Bogomolov, Aleksandra Eliseeva, Timur Galimzyanov, Evgeniy Glukhov, Anton Shapkin, Maria Tigina, Yaroslav Golubev, Alexander Kovrigin, Arie van Deursen, Maliheh Izadi, and Timofey Bryksin. 2024. Long code arena: a set of benchmarks for long-context code models. *Preprint*, arXiv:2406.11612.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Nicolo Cesa-Bianchi, Claudio Gentile, Fabio Vitale, and Giovanni Zappella. 2012. A correlation clustering approach to link classification in signed networks. In *Conference on Learning Theory*, pages 34–1. JMLR Workshop and Conference Proceedings.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Kai-Yang Chiang, Cho-Jui Hsieh, Nagarajan Natarajan, Inderjit S Dhillon, and Ambuj Tewari. 2014. Prediction and clustering in signed networks: a local to global perspective. *The Journal of Machine Learning Research*, 15(1):1177–1213.

S. Chopra, R. Hadsell, and Y. LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yangruibo Ding, Zijian Wang, Wasi Ahmad, Murali Krishna Ramanathan, Ramesh Nallapati, Parminder Bhatia, Dan Roth, and Bing Xiang. 2024. CoCoMIC: Code completion by jointly modeling in-file and cross-file context. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3433–3445, Torino, Italia. ELRA and ICCL.

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. 2020. CodeBERT: A pre-trained model for programming and natural languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1536–1547, Online. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021. Complement lexical retrieval model with semantic residual embeddings. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 – April 1, 2021, Proceedings, Part I*, pages 146–160, Berlin, Heidelberg. Springer-Verlag.

Daya Guo, Shuai Lu, Nan Duan, Yanlin Wang, Ming Zhou, and Jian Yin. 2022. UniXcoder: Unified cross-modal pre-training for code representation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7212–7225, Dublin, Ireland. Association for Computational Linguistics.

Daya Guo, Shuo Ren, Shuai Lu, Zhangyin Feng, Duyu Tang, Shujie LIU, Long Zhou, Nan Duan, Alexey Svyatkovskiy, Shengyu Fu, Michele Tufano, Shao Kun Deng, Colin Clement, Dawn Drain, Neel Sundaresan, Jian Yin, Daxin Jiang, and Ming Zhou. 2021. Graphcode{bert}: Pre-training code representations with data flow. In *International Conference on Learning Representations*.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. 2024. Deepseek-coder: When the large language model meets programming - the rise of code intelligence. *ArXiv*, abs/2401.14196.

Michael Günther, Isabelle Mohr, Daniel James Williams, Bo Wang, and Han Xiao. 2024. Late chunking: Contextual chunk embeddings using long-context embedding models. *Preprint*, arXiv:2409.04701.

Junjie Huang, Duyu Tang, Linjun Shou, Ming Gong, Ke Xu, Daxin Jiang, Ming Zhou, and Nan Duan. 2021. CoSQA: 20,000+ web queries for code search and question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5690–5700, Online. Association for Computational Linguistics.

Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. 2019. Codesearchnet challenge: Evaluating the state of semantic code search. *CoRR*, abs/1909.09436.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. 2024. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*.

Aditya Kanade, Petros Maniatis, Gogul Balakrishnan, and Kensen Shi. 2020. Pre-trained contextual embedding of source code.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA. Association for Computing Machinery.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020a. Pre-training via paraphrasing. In *Advances in Neural Information Processing Systems*, volume 33, pages 18470–18481. Curran Associates, Inc.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020c. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Jia Li, Ge Li, Xuanming Zhang, Yunfei Zhao, Yihong Dong, Zhi Jin, Binhua Li, Fei Huang, and Yongbin Li. 2024a. Evocodebench: An evolving code generation benchmark with domain-specific evaluations. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023. Starcoder: may the source be with you! *Preprint*, arXiv:2305.06161.

Xiangyang Li, Kuicai Dong, Yi Quan Lee, Wei Xia, Yichun Yin, Hao Zhang, Yong Liu, Yasheng Wang, and Ruiming Tang. 2024b. Coir: A comprehensive benchmark for code information retrieval models. *Preprint*, arXiv:2407.02883.

Dianshu Liao, Shidong Pan, Xiaoyu Sun, Xiaoxue Ren, Qing Huang, Zhenchang Xing, Huan Jin, and Qinying Li. 2024. A3-codgen: A repository-level code generation framework for code reuse with local-aware, global-aware, and third-party-library-aware. *Preprint*, arXiv:2312.05772.

Tianyang Liu, Canwen Xu, and Julian McAuley. 2024. Repobench: Benchmarking repository-level code auto-completion systems. In *The Twelfth International Conference on Learning Representations*.

Xiangyan Liu, Bo Lan, Zhiyuan Hu, Yang Liu, Zhicheng Zhang, Fei Wang, Michael Qizhe Shieh, and Wenmeng Zhou. 2025. CodexGraph: Bridging large language models and code repositories via code graph databases. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 142–160, Albuquerque, New Mexico. Association for Computational Linguistics.

Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, Ge Li, Lidong Zhou, Linjun Shou, Long Zhou, Michele Tufano, MING GONG, Ming Zhou, Nan Duan, Neel Sundaresan, Shao Kun Deng, Shengyu Fu, and Shujie LIU. 2021. CodeXGLUE: A machine learning benchmark dataset for code understanding and generation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Yingwei Ma, Qingping Yang, Rongyu Cao, Binhua Li, Fei Huang, and Yongbin Li. 2024. How to understand whole software repository? *ArXiv*, abs/2406.01422.

Siru Ouyang, Wenhao Yu, Kaixin Ma, Zilin Xiao, Zhihan Zhang, Mengzhao Jia, Jiawei Han, Hongming Zhang, and Dong Yu. 2024. Repograph: Enhancing ai software engineering with repository-level code graph. *arXiv preprint arXiv:2410.14684*.

Dragomir R. Radev, Hong Qi, Harris Wu, and Weiguo Fan. 2002. Evaluating web-based question answering systems. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Muhammad Shihab Rashid, Christian Bock, Yuan Zhuang, Alexander Buchholz, Tim Esler, Simon Valentin, Luca Franceschi, Martin Wistuba, Prabhu Teja Sivaprasad, Woo Jung Kim, Anoop Deoras, Giovanni Zappella, and Laurent Callot. 2025. Swe-polybench: A multi-language benchmark for repository level evaluation of coding agents. *Preprint*, arXiv:2504.08703.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with

hard negative samples. In *International Conference on Learning Representations*.

Haifeng Ruan, Yuntong Zhang, and Abhik Roychoudhury. 2024. Specrover: Code intent extraction via llms. *Preprint*, arXiv:2408.02232.

Disha Shrivastava, Denis Kocetkov, Harm de Vries, Dzmitry Bahdanau, and Torsten Scholak. 2023. Repofusion: Training code models to understand your repository. *Preprint*, arXiv:2306.10998.

Kihyuk Sohn. 2016a. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Kihyuk Sohn. 2016b. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Tarun Suresh, Revanth Gangi Reddy, Yifei Xu, Zach Nussbaum, Andriy Mulyar, Brandon Duderstadt, and Heng Ji. 2025. CoRNStack: High-quality contrastive data for better code retrieval and reranking. In *The Thirteenth International Conference on Learning Representations*.

Chongyang Tao, Tao Shen, Shen Gao, Junshuo Zhang, Zhen Li, Zhengwei Tao, and Shuai Ma. 2024. Llms are also effective embedding models: An in-depth overview. *arXiv preprint arXiv:2412.12591*.

Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. *Proceedings of the 19th Australasian Document Computing Symposium*.

Charalampos E Tsourakakis, Michael Mitzenmacher, Kasper Green Larsen, Jarosław Błasiok, Ben Lawson, Preetum Nakkiran, and Vasileios Nakos. 2017. Predicting positive and negative links with noisy queries: Theory & practice. *arXiv preprint arXiv:1709.07308*.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748.

Zora Zhiruo Wang, Akari Asai, Xinyan Velocity Yu, Frank F. Xu, Yiqing Xie, Graham Neubig, and Daniel Fried. 2024. Coderag-bench: Can retrieval augment code generation? *Preprint*, arXiv:2406.14497.

Chunqiu Steven Xia, Yinlin Deng, Soren Dunn, and Lingming Zhang. 2024. Agentless: Demystifying llm-based software engineering agents. *arXiv preprint*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024a. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024b. Swe-agent: Agent-computer interfaces enable automated software engineering. *Preprint*, arXiv:2405.15793.

Dejiao Zhang, Wasi Uddin Ahmad, Ming Tan, Hantian Ding, Ramesh Nallapati, Dan Roth, Xiaofei Ma, and Bing Xiang. 2024a. CODE REPRESENTATION LEARNING AT SCALE. In *The Twelfth International Conference on Learning Representations*.

Fengji Zhang, B. Chen, Yue Zhang, Jin Liu, Daoguang Zan, Yi Mao, Jian-Guang Lou, and Weizhu Chen. 2023a. Repocoder: Repository-level code completion through iterative retrieval and generation. In *Conference on Empirical Methods in Natural Language Processing*.

Fengji Zhang, Bei Chen, Yue Zhang, Jacky Keung, Jin Liu, Daoguang Zan, Yi Mao, Jian-Guang Lou, and Weizhu Chen. 2023b. Repocoder: Repository-level code completion through iterative retrieval and generation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Shuai Zhang, Wang Lijie, Xinyan Xiao, and Hua Wu. 2022. Syntax-guided contrastive learning for pre-trained language model. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2430–2440, Dublin, Ireland. Association for Computational Linguistics.

Yuntong Zhang, Haifeng Ruan, Zhiyu Fan, and Abhik Roychoudhury. 2024b. Autocoderover: Autonomous program improvement. *Preprint*, arXiv:2404.05427.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2024. Large language models for information retrieval: A survey. *Preprint*, arXiv:2308.07107.

Appendix A Related work

Here, we present some works related to ours.

Benchmarks for code retrieval The scope of standard benchmarks for semantic matching or code search between a natural language query and code such as CoSQA (Huang et al., 2021), CodeSearchNet (Husain et al., 2019), AdvTest (Lu et al., 2021), CodeXGLUE (Lu et al., 2021) are based on docstrings and corresponding code. The repository-level retrieval benchmarks such as RepoEval (Zhang et al., 2023a), Repobench (Liu et al., 2024), CodeRAG-Bench (Wang et al., 2024), EvoCodeBench (Li et al., 2024a) are typically used for code completion tasks where they do not have natural language queries or the retrieval is only for context and not editing. CoIR (Li et al., 2024b) is a recent benchmark that curated a collection of multiple datasets for various scenarios, including text-to-code, code-to-text, code-to-code retrieval, thus going beyond our target setting. Therefore, we focus our work on real-world GitHub repositories with natural language queries such as SWE-bench (Jimenez et al., 2024) and Long Code Arena for bug localisation (Bogomolov et al., 2024). CoRNStack (Suresh et al., 2025) is the another dataset that is relevant to us, however this is a concurrent work to ours and we do not experiment with it.

Models for code-editing retrieval Generative Large Language Models (LLMs) are showing improvements on many code related tasks (Li et al., 2023; Guo et al., 2024; Yang et al., 2024a). BehnamGhader et al. (2024) proposes LLM2Vec which allows LLMs to be used for dense retrieval. LLM2Vec comes at a cost of a much higher parameter count for the model model, making it slower during inference, more costly to store embeddings, and challenging to fine-tune. Xia et al. (2024) propose a simple LLM-only framework to code-editing. Their approach prompts the model to localise, repair, and validate their solution. Most current approaches use agents which allows an LLM to interact with the repository through the use of tools such as reading, editing files, and running bash commands (Yang et al., 2024b). A notable LLM-agent AutocodeRover is provided with specific code search APIs which iteratively retrieve code context and locate bugs (Zhang et al., 2024b). Further improvements are seen by SpecRover which generates summaries and feedback messages during agent steps (Ruan et al., 2024). To improve repository-level navigation, Ma et al. (2024) propose an agent RepoUnderstander that condenses the codebase into a knowledge graph and exploit the structure of the repository using Monte Carlo tree search. Liu et al. (2025) parses a repository into code entities and establishes relationships between them through a dataflow analysis, forming a repo-specific context graph. This is shown to improve code completion accuracy. Liu et al. (2025) integrate LLM agents with graph database interfaces extracted from code repositories. By leveraging the structural properties of graph databases and the flexibility of the graph query language, CodexGraph. RepoGraph (Ouyang et al., 2024) constructs a graph of code lines, with the nodes being code lines that capture definition-reference dependencies. These works are orthogonal ways of approaching the code retrieval problem through prompting LLMs.

Representation learning for code Representation learning for programming languages has benefited many downstream applications. Different techniques have been applied to learning representation such as masked language modelling (MLM) (Feng et al., 2020; Li et al., 2023), next token prediction (Kanade et al., 2020; Li et al., 2023) and contrastive learning (Guo et al., 2022; Zhang et al., 2022, 2024a). Representation learning seeks to induce meaningful (vector) embeddings of inputs, and is motivated by applications such as information retrieval via semantic search (Reimers and Gurevych, 2019; Izacard et al., 2021; Zhang et al., 2024a). Successful application of unsupervised contrastive learning leverages text-code pairs, mined from docstrings (Husain et al., 2019; Guo et al., 2022; Zhang et al., 2022, 2024a). These works involve models that semantically align the embeddings of code to its natural language description. However, they do not consider the alignment or abstraction required for retrieval for code-editing queries at the repository-level. Concurrent to our work, CoCoMic (Ding et al., 2024) shows that including relevant cross-file context based on import statements significantly improves retrieval.

Repository-level feature learning The standard loss for code retrieval is contrastive multi-class N-pair loss or InfoNCE (Sohn, 2016b; van den Oord et al., 2018; Chen et al., 2020). This loss maximises similarity between positive pairs while reducing similarity to all other pairs. This loss is

used in many retrieval or semantic search applications (Husain et al., 2019; Karpukhin et al., 2020; Zhang et al., 2024a). Hard-negative mining selects negative examples that differ from the anchor but have similar embeddings, making them the most challenging for the model to distinguish (Robinson et al., 2021). Supervised contrastive learning with hard-negatives has been show to generally improve retrieval performance (Karpukhin et al., 2020; Lewis et al., 2020a,c). However, Xiong et al. (2021) argues that in-batch negatives are unlikely to be hard negatives when the mini-batch size is far smaller than the corpus size and when only a few negatives are informative. This has been shown to produce suboptimal training signals for dense passage retrieval across independent documents. A solution is to use negatives from lexical models such as BM25 during training (Karpukhin et al., 2020; Gao et al., 2021; Luan et al., 2021). Similarly, for code retrieval across independent repositories, we reduce in-batch negatives in favour of sampled hard negatives within the instance repository. Our formulation closely resembles standard maximum likelihood estimation. For each repository, the likelihood function is modeled as a categorical distribution over pairs, each consisting of the query and a repository code chunk. To reduce computational complexity, we approximate the normalisation factor by sampling a number of 'negative' pairs. Our training loss is then computed by averaging across all repositories. On the architecture side, we note that other kinds of fusion like feature fusion have been explored in literature in other contexts (Günther et al., 2024), however for this work, we limit ourselves to input string concatenation for its ease of implementation.

Appendix B Chunking methodology

B.1 Representing file & class information

We form *chunks* of code by splitting each file into its constituent classes, methods and functions as shown in Figure 1.

File hierarchy: These code chunks do not contain any information about their locality in the code repository. To add this information, we insert the file path at the beginning of all code chunks. This is an important bias for code-editing retrieval as the problem statements q's may often contain relevant file paths. For instance, on SWE-bench Verified (Jimenez et al., 2024), 26% of the problem statements contain the path of at least one ground truth file (i.e. one file that needs to be edited) (see Table 4). Empirically we verify the efficacy of this representation by showing improved retrieval performance and visualising the attention maps of a trained retrieval model in Appendix I.

Class representation: Similarly, we preserve the class hierarchy within the chunks. We represent a class by its documentation string, its constructor and declaration of class methods. Each method is represented by including the class it belongs to.

Appendix C Metric Definitions

We measure the performance of our model using standard retrieval metrics: recall@k and mean reciprocal rank (MRR). We report recall at both a file and a chunk level. We compute file recall by retrieving k chunks and take the files those chunks have been extracted from. A similar process is done to the ground truth chunks as well.

$$\text{Recall@}k = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathcal{C}_{F_i} \cap \mathcal{C}_i^*|}{|\mathcal{C}_i^*|} \tag{4}$$

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\text{rank}(\mathcal{C}_i^*, \mathcal{C}_{F_i})} \tag{5}$$

Here rank$(\cdot,\cdot)$ refers to the rank of the first element that is common to both the arguments. This metric measures the minimum value of k needed to retrieve at least one correct code chunk, and is also known as the First Answer Reciprocal Rank (Radev et al., 2002).

Appendix D Dataset statistics

We report relevant dataset statistics in Table 3 and Table 4. If an instance does not have at least one modified function, class or method of class, we discard it for the purpose of computing reported statistics. For all token calculations we use the SentencePiece (SP) tokenizer employed by CodeSage (Zhang et al., 2024a). We extract ground truth (GT) files and chunks using the ground truth patch for each instance, namely a file is in the ground truth if it is edited in the patch, and the same for other chunks. Overall, training instances (SWE-bench train) have slightly larger ground truth sets, comprising on average 5 chunks, than test instances (SWE-bench verified and LCA). File overlap (GT file overlap in Table 4) is overall high across datasets, suggesting that typically changes are restricted to a single or a small number of files.

Finally, more than a quarter of queries in SWE-bench and more than a third in LCA of queries contain the path of at least one file to edit (GT file in query, Table 4).

Appendix E Baseline Model Details

We report in Table 6 various statistics about the models and methods we choose as baselines. Given the trade-offs between the performance and resource requirements, we choose CodeSage S as our base model to build CoRet upon.

Appendix F Baseline performance

We report in Figure 4 the performance of various models and methods on SWE-bench Verified (before fine-tuning).

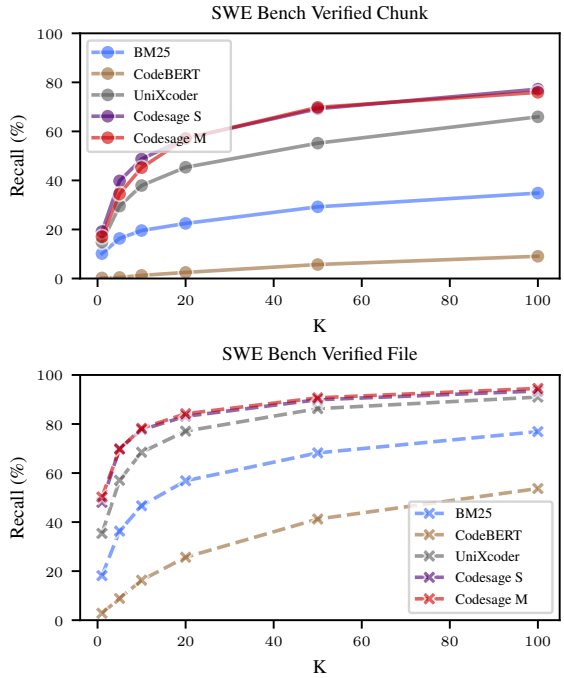


Figure 4: Performance of various models on SWE-bench Verified before fine-tuning. Top: chuck level recall; bottom: file-level recall. It is evident that the modern encoder models like the ones from the CodeSage family perform substantially better than other baselines. However, those models reach acceptable performance (say around 80% chunk recall) only when retrieving a large number of chunks ($k \geq 50$). This motivates our need to train models that are specific for code retrieval.

Appendix G Fine-tuning details

All experiments use the model CodeSage small[1] (Zhang et al., 2024a) for comparability and reproducibility. This Transformer encoder has 6 layers with 8 attention heads per layer. The size for the word embedding vectors and model projections are 1024, feed-forward dimensions are 4096 which leads to models of approximately 130 million trainable parameters. We use bf16-mixed precision to reduce the memory footprint. The model was pre-trained on natural language and code tokenized with a Sentence Piece Tokenizer with a vocabulary of approximately 50K tokens. Each fine-tuning experiment takes approximately 24 hours to run on 8 NVIDIA A10G Tensor Core GPUs. During fine-tuning we use the hyperparameters reported in Table 7.

We choose the optimiser hyperparameters to increase stability and remove the need of initial learning rate warm up. Learning rate was selected over the grid

$$\{1e^{-3}, 5e^{-4}, 1e^{-4}, 5e^{-5}, 1e^{-5}, 5e^{-6}, 1e^{-6}\}$$

The batch size is 1 instance per GPU over 8 GPUs with gradient accumulation of 32; namely, we use an effective batch size of 256 instances. For each instance we sample 1024 (in-repository) negatives during training.

[1]We use the initial version `https://huggingface.co/codesage/codesage-small` as the second version was released after completion of the project.

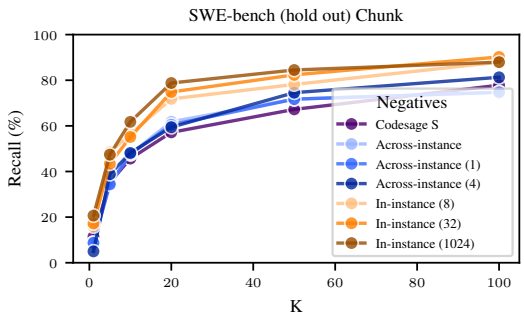


Figure 5: The influence of number of negatives $|\mathcal{B}|$ and their source.

Appendix H Choice of negative samples in Equation (2) influences the performance

In Figure 5, we show the impact of choice of negatives. Karpukhin et al. (2020) propose including hard-negative chunks from BM25 which we select from the same instance – we found marginal improvement when considering a single BM25 negative. In summary, in-instance negatives show a clear advantage, and including more negatives is advantageous for the performance. These results

Dataset		Instances	Files	Chunks	Chunk-level			Query
					GT	Lines	Tokens	Tokens
SWE-bench	Train	8349	$263_{(269)}$	$3628_{(3150)}$	$5_{(18)}$	$20_{(6)}$	$198_{(60)}$	$631_{(1974)}$
SWE-bench	Verified	435	$420_{(222)}$	$8551_{(6118)}$	$2_{(3)}$	$19_{(7)}$	$195_{(84)}$	$539_{(687)}$
LCA	Test	34	$210_{(238)}$	$3243_{(5010)}$	$4_{(5)}$	$19_{(8)}$	$206_{(142)}$	$911_{(878)}$

Table 3: General dataset statistics for SWE-bench dataset. **GT** is the ground truth number of chunks that are edited, **Lines** the average number of lines and **Tokens** the average number of tokens (with SP tokenizer) edited. Standard deviation are reported in brackets.

Dataset		Chunks per File	Files per GT	GT file overlap	GT file in query
SWE-bench	Train	$17_{(8)}$	$1.97_{(3.06)}$	0.83	0.28
SWE-bench	Verified	$22_{(9)}$	$1.24_{(1.06)}$	0.82	0.26
LCA	Test	$16_{(7)}$	$2.4_{(2.24)}$	0.70	0.38

Table 4: **GT file overlap**: On instances where there are multiple chunks to edit, we report the empirical probability that at least one file is in common between the chunks to edit. **GT file in query** is the empirical probability that the query contains at least one file path of the ground truth files to edit.

have been computed on a hold-out set from the SWE-bench train dataset.

Appendix I Repository hierarchy experiment

In this experiment we show that including the file path to the chunks representation is important for the retrieval task and aids in maintaining the repository hierarchy. In Table 2, we show the retrieval performance of models tested with and without including the file path. The BM25 and not-fine-tuned CodeSage model achieve minor improvements in retrieval performance, indicating that fine-tuning is crucial for the model to make use of the added information.

We also investigate to what extent a model trained with file paths rely on them. In Table 2 in the main paper, we show the retrieval performance drop of a model trained with file paths once they are removed. We further analyse this by showing the attention plots of a model trained with file paths when given a query, visualised in Figure 6. For this, we consider a repository from the SWE-bench train subset which was not used in our training set (it is a validation example) and contains the correct file path in the query. We select the second from the last layer of the model and averaged over all attention heads. Figure 6 and Figure 7 shows that the model attends to the file path both for queries and code chunks and highlights that the model has learned to find and leverage the path information in a natural language query.

Appendix J License of artefacts used

SWE-bench (Jimenez et al., 2024) dataset packages multiple repositories that are based on BSD 3-Clause, MIT, Apache-2.0, Custom (based on BSD-2 and BSD-3), GPLv2 licenses. The exact details are available in the original paper in Table 12. The LCA dataset is released under Apache-2, with its constituent data point from repositories with MIT and Apache 2.0 licenses.

The code for training the models for CodeBERT and GraphCodeBERT is released under a MIT license at `https://github.com/microsoft/CodeBERT/blob/master/LICENSE` and their weights are released under with no license specified at `https://huggingface.co/microsoft/codebert-base` and `https://huggingface.co/microsoft/graphcodebert-base` respectively. UniXcoder's weights are released under Apache 2.0 at `https://huggingface.co/microsoft/unixcoder-base`. Our primary baselines, the CodeSage family, have been released under a permissive Apache 2.0 license at `https://huggingface.co/codesage/codesage-small`.

Dataset		**Calls**	**GT call overlap**
SWE-bench	Train	$1.40_{(0.70)}$	0.46
SWE-bench	Verified	$1.92_{(2.22)}$	0.38
SWE-bench	Lite	$2.37_{(2.90)}$	0.16
LCA	Test	$1.15_{(0.49)}$	0.40

Table 5: **GT call overlap**: When there are multiple chunks to edit what is the probability of at least 1 chunk being connected by the call graph. The standard deviation is represented in the brackets.

Model	Parms	Tok	Length	Encoding
BM25	-	words	-	-
CodeBERT	125M	BPE	512	0.5min
GraphCodeBERT	125M	BPE	512	0.5min
UniXcoder	125M	BPE	512	0.5min
CodeSage S	130M	SP	1024	1.5min
CodeSage M	400M	SP	1024	5min
CodeSage L	1.3B	SP	1024	22.5min

Table 6: Statistics about baseline models and methods. Parms: number of model parameters; Tok: tokenizer with SentencePiece (SP), Byte-Pair Encoding (BPE); Length: maximum number input tokens (maximum context length); Encoding the approximate inference time in minutes per 10K chunks, on a single GPU. The model we chose for fine-tuning (CodeSage S) is highlighted. Additionally, the models have been released under permissive licenses, or their official huggingface repositories do not have a license specific.

Hyperparameters	**Full**	**Late Fusion**
Optimiser	RAdam	RAdam
Learning rate	$5e^{-4}$	$5e^{-4}$
Scheduler	cosine decay	cosine decay
Batch size	8	8
Gradient acc.	32	32
Negatives	1024	1024
Epochs	4	8

Table 7: Hyperparameters for fine-tuning

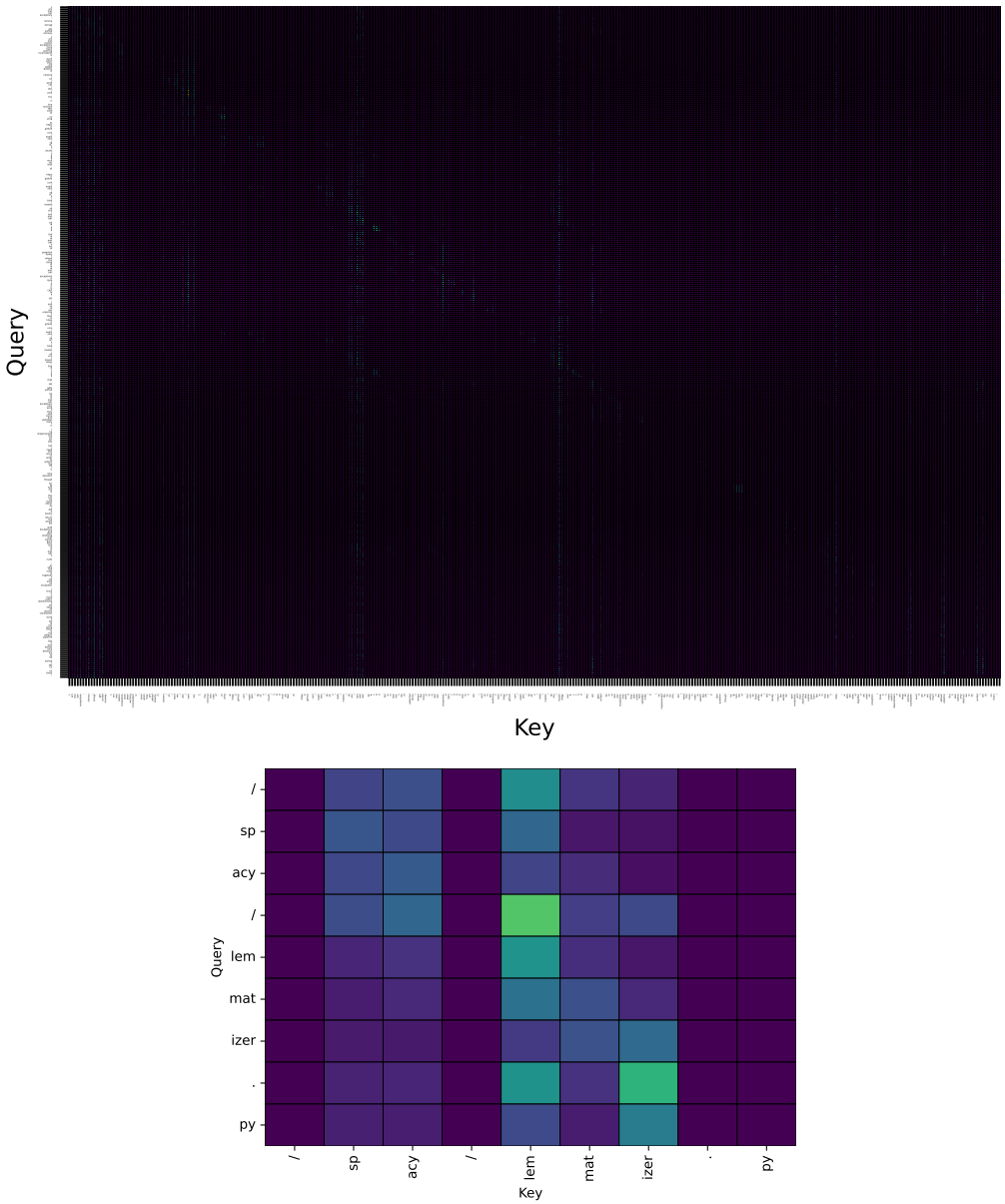


Figure 6: Attention map: query (problem statement) containing file path. Top: full average attention map (best viewed on a screen); bottom: zoom-in portion of the attention map containing the file path. Best viewed in colour on a digital display

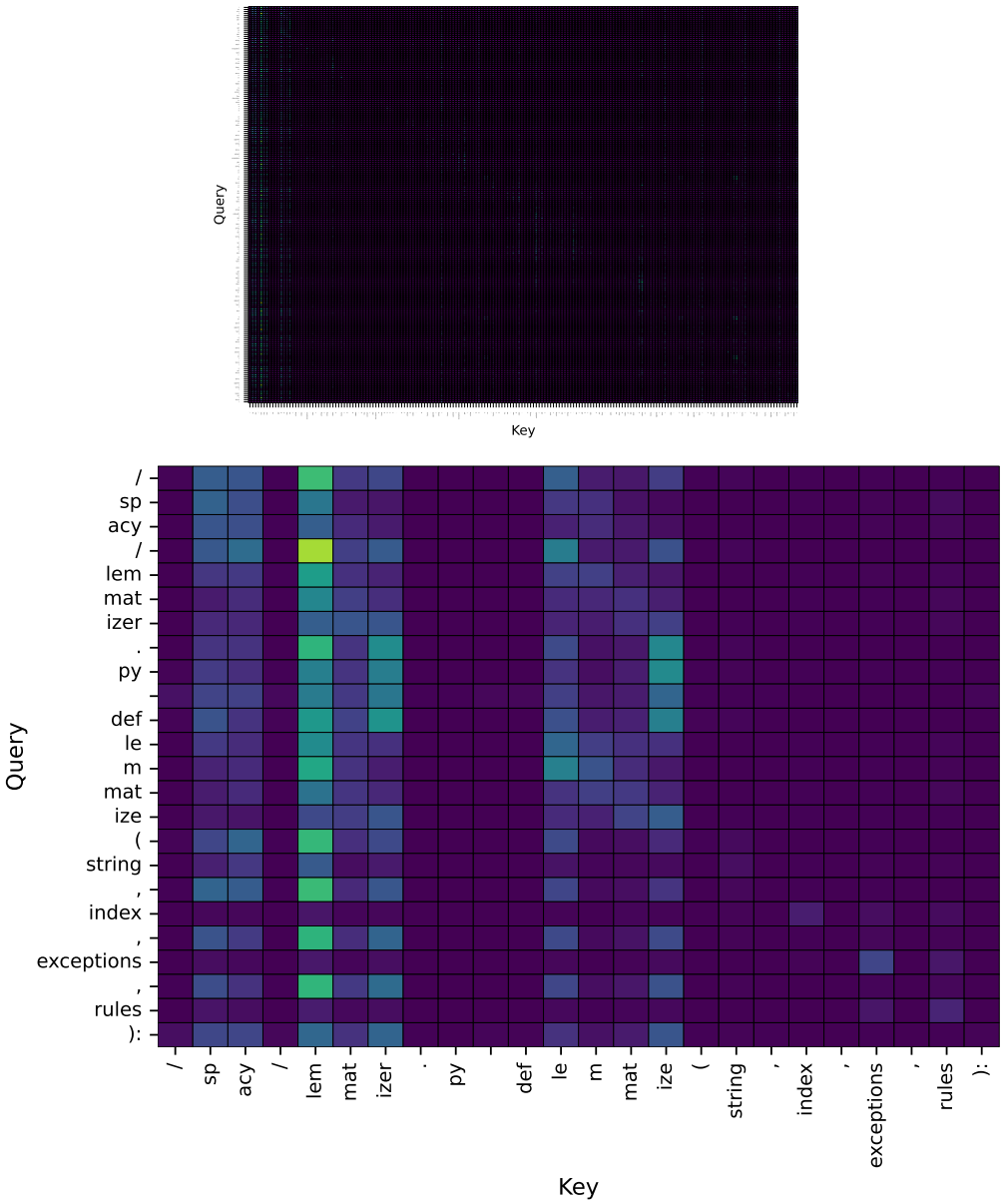


Figure 7: Attention map: code chunk containing file path. Top: full average attention map (best viewed on a screen); bottom: zoom-in portion of the attention map containing the file path. Best viewed in colour on a digital display

Has Machine Translation Evaluation Achieved Human Parity? The Human Reference and the Limits of Progress

Lorenzo Proietti* **Stefano Perrella*** **Roberto Navigli**
Sapienza NLP Group, Sapienza University of Rome
{lproietti, perrella, navigli}@diag.uniroma1.it

Abstract

In Machine Translation (MT) evaluation, metric performance is assessed based on agreement with human judgments. In recent years, automatic metrics have demonstrated increasingly high levels of agreement with humans. To gain a clearer understanding of metric performance and establish an upper bound, we incorporate human baselines in the MT meta-evaluation, that is, the assessment of MT metrics' capabilities. Our results show that human annotators are not consistently superior to automatic metrics, with state-of-the-art metrics often ranking on par with or higher than human baselines. Despite these findings suggesting human parity, we discuss several reasons for caution. Finally, we explore the broader implications of our results for the research field, asking: Can we still reliably measure improvements in MT evaluation? With this work, we aim to shed light on the limits of our ability to measure progress in the field, fostering discussion on an issue that we believe is crucial to the entire MT evaluation community.

1 Introduction and Related Work

Machine Translation (MT) evaluation is the task of assessing the quality of translated text, while MT meta-evaluation estimates the capabilities of automatic evaluation techniques, i.e., MT metrics. Historically, automatic metrics have been employed due to their low cost and fast experimentation time, whereas human evaluation is still considered the gold standard, necessary for validating automatically-derived findings. However, in recent years the MT evaluation field has seen significant advancements. Neural-based metrics have demonstrated strong correlations with human judgments, largely replacing traditional heuristic-based metrics, and becoming the de facto standard in MT evaluation (Freitag et al., 2022, 2023, 2024). More recently, LLM-based approaches to MT evaluation have emerged (Kocmi and Federmann, 2023b,a; Fernandes et al., 2023; Bavaresco et al., 2024), offering not only high correlation with human judgments but also improved interpretability. This raises the question of what is still missing in order for automatic techniques to achieve human parity, if they have not already. Indeed, unlike other Natural Language Processing tasks, MT evaluation lacks a human performance reference, making it difficult to gauge the true capabilities of MT metrics. For instance, in MT, human performance is measured by evaluating human references alongside system translations (Läubli et al., 2018; Kocmi et al., 2023, 2024a). Similarly, popular benchmarks such as HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), and MT-bench (Zheng et al., 2023) report the performance of human baselines.

Since MT metrics' performance is measured based on agreement with human annotators, we posit that agreement among the annotators themselves can serve as a reference for human performance. Previous studies have reported the Inter-Annotator Agreement (IAA) in MT evaluation: Lommel et al. (2014b) used Cohen's kappa to measure the pairwise agreement between raters; Freitag et al. (2021a) grouped raters' assessments into seven score bins before calculating pairwise agreement; and Kocmi et al. (2024b) used Kendall correlation coefficient τ_c. However, these studies employed different measures, making direct comparisons difficult, and none contextualized IAA in relation to the performance of automatic metrics. To the best of our knowledge, Perrella et al. (2024a) were the first to assess metric and human performance jointly. Specifically, they evaluated automatic metrics and human annotators within their new evaluation framework. Nonetheless, since comparing humans and metrics was not their primary focus, they included only one human anno-

*Equal contribution.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 790–813
July 27 - August 1, 2025

	2020		2022		2023		2024
	→ DE	ZH →	→ DE	→ ZH	→ DE	ZH →	→ ES
MQM	3	3	3	3	4	3	1
ESA	✗	✗	✗	✗	2	✗	1
pSQM	3	3	✗	✗	✗	✗	✗
DA+SQM	✗	✗	1	1	1	1	✗
#Seg	681	895	583	1065	145	687	449
#Sys	9	9	10	13	12	15	12

Table 1: The four top rows indicate the number of distinct evaluators for each annotation protocol and test set. We list the studies that released these annotations in Appendix B. '2020' refers to the data released by Freitag et al. (2021a), while other years correspond to the test sets from the corresponding WMT editions. The notation → X indicates that the test set contains translations from English to X, whereas X → denotes translations from X to English. The two bottom rows present the number of source segments and automatic translations per source segment present in the intersection of annotations from all human evaluators, restricted to the segments annotated by disjoint sets of raters (§2.1).

tation protocol – i.e., Direct Assessment + Scalar Quality Metrics (Kocmi et al., 2022a) – that exhibited very poor performance, likely due to low annotation quality, rendering it ineffective as a human performance reference for MT metrics.

In this work, we address this gap by incorporating human baselines into the metric rankings from various editions of the Metrics Shared Task of the Conference on Machine Translation (WMT). By using meta-evaluation strategies from WMT 2024 we derive a single, comprehensive ranking of MT *evaluators* – both human and automatic – establishing a human performance reference for MT metrics across several test sets, translation directions, and human annotation protocols, and offering a clearer understanding of the capabilities of current MT evaluation techniques. Then, given that our results suggest that automatic metrics may have reached human parity, we critically examine this claim and discuss its implications for future research in MT evaluation.[1]

2 Preliminaries and Experimental Setup

In this section, we describe the human annotations, the annotation protocols, the test sets selected for our work, the meta-evaluation strategies employed, and the automatic metrics included.

[1]We publish the code to reproduce our results at https://github.com/SapienzaNLP/human-parity-mt-eval.

2.1 The Human Annotations

Each year, WMT conducts manual annotation campaigns to collect human judgments of translation quality. First, each test set t is created by drawing N_t segments from various sources. Segments may consist of individual sentences or entire paragraphs. Each source segment is then translated into the target language using M_t MT systems, producing $N_t \times M_t$ translations per test set t. Finally, human raters assess translation quality (Kocmi et al., 2023, 2024a; Freitag et al., 2023, 2024).

Given the large volume of translations, non-overlapping portions of each test set are typically assigned to different raters. Consequently, the annotated test sets used in this work combine annotations from multiple raters. For simplicity, we use the term *evaluator* to refer to any entity that produced a set of annotations covering all segments in a test set. An evaluator can be a human rater, an MT metric, an ensemble of MT metrics, or an entity that selects annotations from different raters. For example, in the test set that we dub "2020 EN→DE" (Freitag et al., 2021a), six raters provided a total of three annotations per translation, yielding three distinct evaluators.

However, this setup introduces a problem: Distinct human evaluators may be derived from non-disjoint sets of raters. If the same rater contributes to multiple evaluators, even across non-overlapping segments, it can artificially inflate their agreement and overestimate human baseline performance. To avoid this, we restrict each test set to the largest subset of segments annotated by strictly disjoint sets of raters. Returning to the 2020 EN→DE example, we aim to partition the six raters into three groups, so that the combined annotations from raters within each group form a single evaluator. Yet, two factors prevent such a simple partitioning: i) not all raters annotated every source segment, and ii) the specific rater-to-segment assignment prevents partitioning raters such that the combined annotations from each group cover all segments. Therefore, we restrict our test set to the segments that allow such a partitioning by solving the following optimization problem: *Find the largest subset of segments and a partitioning of raters into three disjoint groups such that each group cumulatively annotated the entire subset of segments*. We apply a similar procedure to each test set with annotations of this form, reporting resulting test set sizes in Table 1. Further details are provided in Appendix A.

Test set 2020	EN→DE				ZH→EN			
	SPA		acc^*_{eq}		SPA		acc^*_{eq}	
Metric	Rank	Acc.	Rank	Acc.	Rank	Acc.	Rank	Acc.
MQM-2020-2	1	96.45	1	58.86	1	88.10	1	55.70
pSQM-1	1	95.59	6	49.41	1	79.16	13	43.89
MQM-2020-3	2	90.39	2	56.84	1	92.06	2	52.80
BLEURT-0.2	2	86.81	4	50.81	2	72.59	3	50.57
pSQM-2	2	85.87	9	46.97	1	89.33	9	46.77
BLEURT-20	2	85.52	3	51.68	3	67.46	4	50.12

Test set 2022	EN→DE				EN→ZH			
	SPA		acc^*_{eq}		SPA		acc^*_{eq}	
Metric	Rank	Acc.	Rank	Acc.	Rank	Acc.	Rank	Acc.
MetricX-23-QE-XXL*	1	94.89	3	57.64	2	83.92	2	47.43
MQM-2022-2	1	94.49	6	55.55	2	80.82	3	47.05
MQM-2022-3	1	92.59	1	61.06	1	87.22	2	47.56
MetricX-23-XXL	2	92.34	2	59.27	1	87.69	1	48.43
DA+SQM	6	66.61	16	46.03	2	82.95	12	36.26

Test set 2023	EN→DE				ZH→EN			
	SPA		acc^*_{eq}		SPA		acc^*_{eq}	
Metric	Rank	Acc.	Rank	Acc.	Rank	Acc.	Rank	Acc.
GEMBA-MQM*	1	94.52	5	58.52	1	93.17	3	52.80
MQM-2023-3	1	93.51	5	58.42	1	95.54	5	51.65
MQM-2023-2	1	93.15	6	57.71	1	95.18	2	52.90
XCOMET-Ensemble	1	92.21	3	60.99	2	91.15	1	54.59
MetricX-23-QE-XXL*	1	92.12	1	62.53	3	88.30	2	53.26
DA+SQM	2	91.24	14	46.79	4	86.28	22	39.42
ESA-1	2	90.39	14	46.71	–	–	–	–
ESA-2	2	89.11	12	49.70	–	–	–	–
MQM-2023-4	2	88.93	14	46.68	–	–	–	–

Test set 2024	EN→ES			
	SPA		acc^*_{eq}	
Metric	Rank	Acc.	Rank	Acc.
CometKiwi-XXL*	1	86.12	4	67.24
gemba_esa*	1	85.72	3	67.68
ESA	2	80.12	8	63.84
metametrics_mt_mqm_hybrid_kendall	2	80.10	1	68.95
MetricX-24-Hybrid	2	79.75	1	69.20

Table 2: Results obtained by applying the WMT 2024 Meta-Evaluation strategies to the test sets illustrated in Section 2.2. The 'Acc.' column contains the Meta-Evaluation accuracy, while 'Rank' reports clusters of statistical significance computed following Freitag et al. (2024), using the PERM-BOTH hypothesis test introduced by Deutsch et al. (2021). Starred metrics are reference-less metrics, and rows highlighted in gray are human evaluators.

2.2 Test Sets and Annotation Protocols

We estimate human performance based on the agreement among human evaluators. Specifically, we designate one human evaluator as ground truth while the others serve as human baselines. Consequently, our setup necessitates multiple human annotations for the same translations. Test sets satisfying this requirement include those released by Freitag et al. (2021a) and those from WMT editions between 2022 and 2024.

These test sets feature human annotations from at least two of the following protocols: Multidimensional Quality Metrics (MQM, Lommel et al., 2014a), Error Span Annotation (ESA, Kocmi et al., 2024b), Professional Scalar Quality Metrics (pSQM, Freitag et al., 2021a), and Direct Assess-

ments + Scalar Quality Metrics (DA+SQM, Kocmi et al., 2022a). Our work leverages these test sets, but we restrict them to source segments that simultaneously: i) were annotated by all considered human evaluators and ii) were annotated by disjoint sets of raters (as detailed in Section 2.1). Table 1 presents statistics for the test sets employed. Additionally, we illustrate all the considered annotation protocols in Appendix B.

Following standard practice in the literature (Freitag et al., 2021a,b, 2022, 2023, 2024), we designate evaluators derived from the MQM annotations released annually at WMT as the ground truth, employing the others as human baselines. Indeed, the MQM protocol relies on experienced annotators and provides a more detailed (and more expensive) evaluation compared to other protocols. Nonetheless, in Appendix F, we also investigate the effects of selecting alternative evaluators – either MQM evaluators different from those previously used or evaluators following different protocols – as the ground truth.

2.3 The MT Meta-Evaluation

We compute metric rankings using the meta-evaluation strategies employed at the WMT 2024 Metrics Shared Task:

- **Soft Pairwise Accuracy (SPA)** estimates evaluator performance based on the ability to rank *MT systems*[2] in the same order as in the ranking derived from ground truth annotations (Thompson et al., 2024).
- **Pairwise Accuracy with Tie Calibration ($\mathbf{acc}^*_{eq}$)** estimates evaluator performance based on the ability to rank *translations of the same source segment* in the same order as in the ranking derived from ground truth annotations (Deutsch et al., 2023).

We describe these measures in more detail in Appendix C.

2.4 Metrics

The automatic evaluators considered – i.e., the MT metrics – are those submitted to the WMT Metrics Shared Task in the 2020, 2022, 2023, and 2024 editions. Additionally, we include several state-of-the-art metrics from recent WMT editions in rankings from previous years, provided they were not trained on the corresponding test sets. Table 3 in Appendix D lists all considered metrics.

[2]The score assigned to an MT system is the average of the scores given to its translations.

3 Results

Table 2 presents the evaluator rankings. Due to space constraints, each table includes only a subset of evaluators. A complete set of results, including all the evaluators, is provided in Appendix E.

Results vary across years and translation directions. Notably, human evaluators do not consistently rank higher than automatic metrics. Under SPA, human evaluators often share clusters of statistical significance with automatic metrics, whereas, under acc^*_{eq}, they are frequently surpassed. For example, in 2020 EN→DE, BLEURT-0.2 and BLEURT-20 fall within the same statistical significance cluster as MQM-2020-3 and pSQM-2 under SPA, with pSQM-2 ranking as low as 9th under acc^*_{eq}. Similarly, in 2022 EN→DE, MQM-2022-2 and MQM-2022-3 share the top cluster with MetricX-23-QE-XXL under SPA, with MQM-2022-2 ranking 6th under acc^*_{eq}. Finally, in 2023 and 2024, most human evaluators rank consistently below various automatic metrics under both SPA and acc^*_{eq}. Even when restricted to the human evaluators who follow the same protocol as the annotations employed as gold – i.e., MQM – they rank consistently in the top positions solely in 2020. Additionally, our findings remain valid when varying the human evaluators used as ground truth, as shown in Appendix F.

These results may indicate human-level performance in MT evaluation. Nonetheless, we argue that they are insufficient to establish equivalence between human and automatic evaluators, and elaborate our reasons in the next section.

4 Discussion

In the same spirit as Tedeschi et al. (2023), who discuss the meaning of superhuman performance in Natural Language Understanding, we outline several factors to consider before making similar claims in MT evaluation. We then discuss the broader implications of our findings, warning that measuring progress in the field may become increasingly challenging.

Meta-evaluation Certain meta-evaluation measures may be inadequate for comparing human and automatic evaluators. In particular, our results consistently rank human evaluators much lower under

acc^*_{eq} than under SPA. This discrepancy may be related to the findings of Perrella et al. (2024b), who show that acc^*_{eq} favors evaluators whose assessments fall within a continuous interval, whereas, as detailed in Appendix B, human evaluators produce discrete assessments.

Annotation quality Certain annotation campaigns might have produced low-quality annotations, either due to a lack of clarity in the annotation guidelines or to the ability of the raters involved. This is particularly concerning in the 2023 EN→DE test set, where, even if restricted to SPA, most human evaluators fall within the second cluster of statistical significance, alongside surface-level metrics such as BLEU.[3]

Benchmarks difficulty Current test sets might be too easy for the MT systems whose translations are being evaluated. Supporting this hypothesis, we observe that sentinel-cand-mqm, a metric that assesses only translation fluency, ranks on par with the human evaluator ESA under SPA, and even higher under acc^*_{eq} (Table 7). This suggests that the evaluated translations may differ only in minor fluency-related nuances. Arguably, to assess whether human parity has been truly achieved, future studies should compare metrics and humans in more demanding contexts. Indeed, previous research has shown that metrics may struggle in unseen domains (Zouhar et al., 2024) and lack sensitivity to specific translation errors such as incorrect number, gender (Karpinska et al., 2022), or word sense disambiguation (Martelli et al., 2025).

4.1 Can We Still Measure Improvements in MT Evaluation?

As discussed, we believe claiming human parity is premature without first addressing the issues outlined above. Nonetheless, with automatic metrics ranking the same as, or higher than, human evaluators in standard benchmarks, our results raise a critical concern about our ability to measure progress in MT evaluation: What does a higher or lower ranking truly mean?

If a metric ranks higher than a human evaluator using a non-MQM protocol, is the metric a better evaluator, or does it merely align more closely with the score distribution of the MQM protocol? More concerningly, if a metric ranks higher than an MQM evaluator, does this suggest superior evaluation capabilities, or does it simply reflect better alignment with the specific raters who produced the gold annotations? Indeed, Finkelstein et al. (2024) achieved an exceptionally high agreement with gold annotations by explicitly optimizing their metric to align with the raters themselves. More generally, we argue that in current benchmarks it is unclear whether a higher ranking – relative to either a human or an automatic evaluator – reflects genuine improvements in evaluation quality or merely closer alignment with a particular annotation protocol or rater style.

To ensure the reliability of meta-evaluation, future research should focus on exploring whether the gap between human and automatic evaluators can be restored. This could be pursued in several ways, including (but not limited to) selecting more challenging test sets, using test sets adversarial to MT metrics (e.g., from domains different from their training data), producing higher-quality human annotations, or designing new annotation protocols that yield stronger inter-annotator agreement. Additionally, greater resources could be allocated to human annotation campaigns – either by collecting multiple annotations per translation to reach a consensus among annotators or by increasing the number of segments in test sets, as suggested by Riley et al. (2024).

5 Conclusions

We incorporate human baselines into the metric rankings from previous editions of the WMT Metrics Shared Task. Our results show that MT metrics frequently rank higher than human evaluators, particularly when the latter follow annotation protocols different from MQM – the protocol used as the ground truth. While our findings suggest that metrics may have reached human-level performance, we recommend caution and highlight several issues the research community should address to assess whether human parity has been truly achieved. Finally, we discuss a critical concern arising from our findings: the limits of measuring progress in MT evaluation as automatic metrics approach human baselines. In this respect, we propose research directions to ensure that progress remains measurable or, at the very least, to extend the period during which it can be reliably tracked.

[3]We wish to highlight that our 2023 test set features only 145 segments annotated by all human evaluators (as reported in Table 1), which might have resulted in unreliable estimates of SPA and acc^*_{eq}.

Limitations

This study required test sets annotated by multiple human evaluators. Consequently, our analysis is limited to seven test sets including four language directions.

Moreover, assessing the agreement between various human evaluators required restricting our analysis to segments annotated by all of them. As a result, some test sets contain only a small number of segments, which might reduce the reliability of the results. To mitigate this issue, our findings are supported by statistical significance analyses.

Acknowledgements

We gratefully acknowledge the support of the PNRR MUR project PE0000013-FAIR, and the CREATIVE project (CRoss-modal understanding and gEnerATIon of Visual and tExtual content), which is funded by the MUR Progetti di Rilevante Interesse Nazionale programme (PRIN 2020). The authors acknowledge the CINECA award IsCb9_mtmit under the ISCRA initiative for the availability of high-performance computing resources.

This work was carried out while Lorenzo Proietti was enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome. The authors thank Anna Bavaresco for her valuable feedback on the manuscript.

References

David Anugraha, Garry Kuwanto, Lucky Susanto, Derry Tanti Wijaya, and Genta Winata. 2024. MetaMetrics-MT: Tuning meta-metrics for machine translation via human preference calibration. In *Proceedings of the Ninth Conference on Machine Translation*, pages 459–469, Miami, Florida, USA. Association for Computational Linguistics.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, André F. T. Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2024. Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks. *Preprint*, arXiv:2406.18403.

Rachel Bawden, Biao Zhang, Andre Tättar, and Matt Post. 2020. ParBLEU: Augmenting metrics with automatic paraphrases for the WMT'20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 887–894, Online. Association for Computational Linguistics.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2021. A statistical analysis of summarization evaluation metrics using resampling methods. *Transactions of the Association for Computational Linguistics*, 9:1132–1146.

Daniel Deutsch, George Foster, and Markus Freitag. 2023. Ties matter: Meta-evaluating modern metrics with pairwise accuracy and tie calibration. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12914–12929, Singapore. Association for Computational Linguistics.

Sören Dreano, Derek Molloy, and Noel Murphy. 2023a. Embed_Llama: Using LLM embeddings for the metrics shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 738–745, Singapore. Association for Computational Linguistics.

Sören Dreano, Derek Molloy, and Noel Murphy. 2023b. Tokengram_F, a fast and accurate token-based chrF++ derivative. In *Proceedings of the Eighth Conference on Machine Translation*, pages 730–737, Singapore. Association for Computational Linguistics.

Muhammad ElNokrashy and Tom Kocmi. 2023. eBLEU: Unexpectedly good machine translation evaluation using simple word embeddings. In *Proceedings of the Eighth Conference on Machine Translation*, pages 746–750, Singapore. Association for Computational Linguistics.

Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André Martins, Graham Neubig, Ankush Garg, Jonathan Clark, Markus Freitag, and Orhan Firat. 2023. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1066–1083, Singapore. Association for Computational Linguistics.

Mara Finkelstein, Dan Deutsch, Parker Riley, Juraj Juraska, Geza Kovacs, and Markus Freitag. 2024. From jack of all trades to master of one: Specializing llm-based autoraters to a test set. *Preprint*, arXiv:2411.15387.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Nitika Mathur, Daniel Deutsch, Chi-Kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian

Thompson, Frederic Blain, Tom Kocmi, Jiayi Wang, David Ifeoluwa Adelani, Marianna Buchicchio, Chrysoula Zerva, and Alon Lavie. 2024. Are LLMs breaking MT metrics? results of the WMT24 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 47–81, Miami, Florida, USA. Association for Computational Linguistics.

Markus Freitag, Nitika Mathur, Chi-kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Tom Kocmi, Frederic Blain, Daniel Deutsch, Craig Stewart, Chrysoula Zerva, Sheila Castilho, Alon Lavie, and George Foster. 2023. Results of WMT23 metrics shared task: Metrics might be guilty but references are not innocent. In *Proceedings of the Eighth Conference on Machine Translation*, pages 578–628, Singapore. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Thamme Gowda, Tom Kocmi, and Marcin Junczys-Dowmunt. 2023. Cometoid: Distilling strong reference-based machine translation metrics into Even stronger quality estimation metrics. In *Proceedings of the Eighth Conference on Machine Translation*, pages 751–755, Singapore. Association for Computational Linguistics.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. MetricX-24: The Google submission to the WMT 2024 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 492–504, Miami, Florida, USA. Association for Computational Linguistics.

Juraj Juraska, Mara Finkelstein, Daniel Deutsch, Aditya Siddhant, Mehdi Mirzazadeh, and Markus Freitag. 2023. MetricX-23: The Google submission to the WMT 2023 metrics shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 756–767, Singapore. Association for Computational Linguistics.

Marzena Karpinska, Nishant Raj, Katherine Thai, Yixiao Song, Ankita Gupta, and Mohit Iyyer. 2022. DEMETR: Diagnosing evaluation metrics for translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9540–9561, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, Mariya Shmatova, Steinthór Steingrímsson, and Vilém Zouhar. 2024a. Findings of the WMT24 general machine translation shared task: The LLM era is here but MT is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46, Miami, Florida, USA. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022a. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023a. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775, Singapore. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023b. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Tom Kocmi, Hitokazu Matsushita, and Christian Federmann. 2022b. MS-COMET: More and better human judgements improve metric performance. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 541–548, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Tom Kocmi, Vilém Zouhar, Eleftherios Avramidis, Roman Grundkiewicz, Marzena Karpinska, Maja Popović, Mrinmaya Sachan, and Mariya Shmatova. 2024b. Error span annotation: A balanced approach for human evaluation of machine translation. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1440–1453, Miami, Florida, USA. Association for Computational Linguistics.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Yilun Liu, Xiaosong Qiao, Zhanglin Wu, Su Chang, Min Zhang, Yanqing Zhao, Song Peng, Shimin Tao, Hao Yang, Ying Qin, Jiaxin Guo, Minghan Wang, Yinglu Li, Peng Li, and Xiaofeng Zhao. 2022. Partial could be better than whole. HW-TSC 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 549–557, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014a. Multidimensional quality metrics (mqm) : A framework for declaring and describing translation quality metrics. *Tradumàtica*, pages 455–463.

Arle Richard Lommel, Maja Popovic, and Aljoscha Burchardt. 2014b. Assessing inter-annotator agreement for translation error annotation. In *MTE: Workshop on Automatic and Manual Metrics for Operational Translation Evaluation. International Conference on Language Resources and Evaluation (LREC-14), located at LREC 14, May 26-31, Reykjavik, Iceland*. LREC.

Federico Martelli, Stefano Perrella, Niccolò Campolungo, Tina Munda, Svetla Koeva, Carole Tiberius, and Roberto Navigli. 2025. Dibimt: A gold evaluation benchmark for studying lexical ambiguity in machine translation. *Computational Linguistics*, pages 1–71.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2799–2808, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725, Online. Association for Computational Linguistics.

Ananya Mukherjee, Hema Ala, Manish Shrivastava, and Dipti Misra Sharma. 2020. Mee : An automatic metric for evaluation using embeddings for machine translation. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 292–299.

Ananya Mukherjee and Manish Shrivastava. 2022a. REUSE: REference-free UnSupervised quality estimation metric. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 564–568, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ananya Mukherjee and Manish Shrivastava. 2022b. Unsupervised embedding-based metric for MT evaluation with improved human correlation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 558–563, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ananya Mukherjee and Manish Shrivastava. 2023. MEE4 and XLsim : IIIT HYD‘s submissions’ for WMT23 metrics shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 800–805, Singapore. Association for Computational Linguistics.

Ananya Mukherjee and Manish Shrivastava. 2024. chrF-S: Semantics is all you need. In *Proceedings of the Ninth Conference on Machine Translation*, pages 470–474, Miami, Florida, USA. Association for Computational Linguistics.

Subhajit Naskar, Daniel Deutsch, and Markus Freitag. 2023. Quality estimation using minimum Bayes risk. In *Proceedings of the Eighth Conference on Machine Translation*, pages 806–811, Singapore. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Stefano Perrella, Lorenzo Proietti, Pere-Lluís Huguet Cabot, Edoardo Barba, and Roberto Navigli. 2024a. Beyond correlation: Interpretable evaluation of machine translation metrics. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20689–20714, Miami, Florida, USA. Association for Computational Linguistics.

Stefano Perrella, Lorenzo Proietti, Alessandro Scirè, Edoardo Barba, and Roberto Navigli. 2024b. Guardians of the machine translation meta-evaluation: Sentinel metrics fall in! In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16216–16244, Bangkok, Thailand. Association for Computational Linguistics.

Stefano Perrella, Lorenzo Proietti, Alessandro Scirè, Niccolò Campolungo, and Roberto Navigli. 2022. MaTESe: Machine translation evaluation as a sequence tagging problem. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 569–577, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022a. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are references really needed? unbabel-IST 2021 submission for the metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online. Association for Computational Linguistics.

Ricardo Rei, Nuno M. Guerreiro, JosÃ© Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de Souza, and André Martins. 2023. Scaling up CometKiwi: Unbabel-IST 2023 submission for the quality estimation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 841–848, Singapore. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020a. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020b. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 911–920, Online. Association for Computational Linguistics.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022b. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Parker Riley, Daniel Deutsch, George Foster, Viresh Ratnakar, Ali Dabirmoghaddam, and Markus Freitag. 2024. Finding replicable human evaluations via stable ranking probability. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4908–4919, Mexico City, Mexico. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020a. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Thibault Sellam, Amy Pu, Hyung Won Chung, Sebastian Gehrmann, Qijun Tan, Markus Freitag, Dipanjan Das, and Ankur Parikh. 2020b. Learning to evaluate translation beyond English: BLEURT submissions to the WMT metrics 2020 shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 921–927, Online. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Peter Stanchev, Weiyue Wang, and Hermann Ney. 2019. EED: Extended edit distance measure for machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 514–520, Florence, Italy. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Simone Tedeschi, Johan Bos, Thierry Declerck, Jan Hajič, Daniel Hershcovich, Eduard Hovy, Alexander Koller, Simon Krek, Steven Schockaert, Rico Sennrich, Ekaterina Shutova, and Roberto Navigli. 2023. What‘s the meaning of superhuman performance in today‘s NLU? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12471–12491, Toronto, Canada. Association for Computational Linguistics.

Brian Thompson, Nitika Mathur, Daniel Deutsch, and Huda Khayrallah. 2024. Improving statistical significance in human evaluation of automatic metrics via soft pairwise accuracy. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1222–1234, Miami, Florida, USA. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020a. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020b. Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity. In *Proceedings of the Fifth Conference on Machine Translation*, pages 561–570, Online. Association for Computational Linguistics.

Giorgos Vernikos, Brian Thompson, Prashant Mathur, and Marcello Federico. 2022. Embarrassingly easy document-level MT metrics: How to convert any pretrained metric into a document-level metric. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 118–128, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Vasiliy Viskov, George Kokush, Daniil Larionov, Steffen Eger, and Alexander Panchenko. 2023. Semantically-informed regressive encoder score. In *Proceedings of the Eighth Conference on Machine Translation*, pages 815–821, Singapore. Association for Computational Linguistics.

Yu Wan, Keqin Bao, Dayiheng Liu, Baosong Yang, Derek F. Wong, Lidia S. Chao, Wenqiang Lei, and Jun Xie. 2022a. Alibaba-translate China‘s submission for WMT2022 metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 586–592, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Yu Wan, Dayiheng Liu, Baosong Yang, Haibo Zhang, Boxing Chen, Derek Wong, and Lidia Chao. 2022b. UniTE: Unified translation evaluation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8117–8127, Dublin, Ireland. Association for Computational Linguistics.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTer: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 505–510, Berlin, Germany. Association for Computational Linguistics.

Zhanglin Wu, Yilun Liu, Min Zhang, Xiaofeng Zhao, Junhao Zhu, Ming Zhu, Xiaosong Qiao, Jingfei Zhang, Ma Miaomiao, Zhao Yanqing, Song Peng, Shimin Tao, Hao Yang, and Yanfei Jiang. 2023. Empowering a metric with LLM-assisted named entity annotation: HW-TSC‘s submission to the WMT23 metrics shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 822–828, Singapore. Association for Computational Linguistics.

Jin Xu, Yinuo Guo, and Junfeng Hu. 2020. Incorporate semantic structures into machine translation evaluation via UCCA. In *Proceedings of the Fifth Conference on Machine Translation*, pages 934–939, Online. Association for Computational Linguistics.

Wenda Xu, Xian Qian, Mingxuan Wang, Lei Li, and William Yang Wang. 2023. SESCORE2: Learning text generation evaluation via synthesizing realistic mistakes. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5166–5183, Toronto, Canada. Association for Computational Linguistics.

Wenda Xu, Yi-Lin Tuan, Yujie Lu, Michael Saxon, Lei Li, and William Yang Wang. 2022. Not all errors are equal: Learning text generation metrics using stratified error synthesis. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6559–6574, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *Proceedings of the International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Vilém Zouhar, Shuoyang Ding, Anna Currey, Tatyana Badeka, Jenyuan Wang, and Brian Thompson. 2024. Fine-tuned machine translation metrics struggle in unseen domains. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 488–500, Bangkok, Thailand. Association for Computational Linguistics.

A Fair Extraction of Evaluators from Human Annotations

The human evaluation campaigns conducted by Freitag et al. (2021a), Freitag et al. (2023), and Riley et al. (2024) produced multiple annotations for each translation. As you can see in Table 1, there are many annotations per translation for MQM and pSQM in the test sets 2020, 2022, and 2023. As discussed in Section 2.1, these annotation campaigns distributed the annotation workload among multiple raters.

Since we derive multiple evaluators from these annotations (some used as ground truth and some as human baselines), we prevent artificially inflating their agreement by not allowing the same rater to contribute to two distinct evaluators simultaneously. For example, in the 2020 EN→DE test set, six raters provided a total of three annotations per translation. We extract three human evaluators from these annotations, using one as the ground truth and the other two as human evaluators (MQM-2020-2 and MQM-2020-3 in Table 2). To achieve this, we partition the six raters into three groups, each forming one evaluator. However, not all raters annotated the entire set of source segments, and the distribution of workload did not allow for a partition that covered all annotated segments. Therefore, to retain the maximum number of segments in our test sets, we solved the following optimization problem: *Find the largest subset of segments and a partitioning of raters into three disjoint groups such that each group cumulatively annotated the entire subset of segments*.

Formally, let us define a test set $t = \{s_1, ..., s_{N_t}\}$ as a set of N_t segments. Each segment was annotated by k out of R raters, with $\mathcal{R} = \{r_1, ..., r_R\}$ representing the set of raters. Our goal is to determine a partition $\Pi = \{\mathcal{R}_1, ..., \mathcal{R}_k\}$ of $\mathcal{R}$ and a subset $u \subseteq t$ such that u is the largest subset in which every segment has been annotated by exactly one rater from each of the k sets in the partition Π.

To solve this optimization problem, we formulate it as an Integer Linear Programming (ILP) problem and solve it using the PuLP[4] Python library. We applied this procedure to the 2020, 2022, and 2023 test sets.

B Human Annotations

We briefly illustrate how each annotation protocol considered works:

- Multidimensional Quality Metrics (MQM) requires annotators to identify error spans in the translated text, specifying error category and severity, to be selected among Neutral, Minor, Major, and Critical. A translation quality score is derived by assigning a penalty to each error span depending on severity (Lommel et al., 2014a; Freitag et al., 2021a).
- Error Span Annotation (ESA) requires annotators to identify error spans in the translated text, specify error severity, and later assign a scalar quality score from 0 to 100 to the translation (Kocmi et al., 2024b).
- Scalar Quality Metrics (SQM) requires annotators to assign a scalar quality score from 0 to 6 to the translated text. Following (Freitag et al., 2021a), we use 'pSQM' to refer to SQM conducted by professional annotators.[5]

[4] https://coin-or.github.io/pulp/.

[5] In this work, we use only annotations produced by professional annotators or translators. Therefore, we exclude cSQM and Direct Assessments (DA) – which were crowdsourced – from the 2020 test sets.

- Direct Assessments + Scalar Quality Metrics (Kocmi et al., 2022a, DA+SQM) requires raters to assign a scalar quality score from 0 to 100 to the translated text. Raters are presented with seven labeled tick marks describing translation quality levels at various score thresholds, similarly to the SQM protocol.

Here, for each set of annotations employed in this work (i.e., those reported in Table 1), we indicate the reference paper that released them:

- The MQM-based and pSQM-based annotations for the test sets 2020 EN→DE and 2020 ZH→EN have been released by Freitag et al. (2021a).
- The MQM-based annotations for the test sets 2022 EN→DE and 2022 EN→ZH have been released by Freitag et al. (2022) and Riley et al. (2024).
- The DA+SQM-based annotations for the test sets 2022 EN→DE and 2022 EN→ZH have been released by Kocmi et al. (2022a).
- Three sets of MQM-based annotations for the test sets 2023 EN→DE and ZH→EN have been released by Freitag et al. (2023).
- The ESA-based annotations and the last set of MQM-based annotations (MQM-2023-4 in Table 2) for the test set 2023 EN→DE have been released by Kocmi et al. (2024b).
- The ESA-based annotations for the test set 2024 EN→ES have been released by Kocmi et al. (2024a).
- The MQM-based annotations for the test set 2024 EN→ES have been released by Freitag et al. (2024).

C Meta-Evaluation Measures

In this section, we describe the two meta-evaluation measures used in our work, as listed in Section 2.3.

C.1 Soft Pairwise Accuracy (SPA)

Thompson et al. (2024) introduced Soft Pairwise Accuracy (SPA) as an extension of Pairwise Accuracy (Kocmi et al., 2021, PA).

Given a test set t, which consists of N_t source segments and M_t translations generated by the respective M_t MT systems (as described in Section 2.1), PA counts how often an evaluator e ranks system pairs in the same order as the ground truth g. Let a_{ij} be 1 if evaluator e ranks systems i and j in the same order as the ground truth and 0 otherwise, where $i, j \in \{0, ..., Mt\}$. Then, PA is defined as:

$$PA = \binom{N}{2}^{-1} \sum_{i=0}^{M_t} \sum_{j=i+1}^{M_t} a_{ij} \quad (1)$$

SPA extends PA by incorporating the confidence with which an evaluator and the ground truth rank two MT systems. Confidence is represented using statistical p-values. Specifically, p^e_{ij} denotes the p-value associated with the statistical hypothesis that system i is better than system j according to evaluator e, while p^g_{ij} represents the corresponding p-value for the ground truth g. SPA is then defined as follows:

$$SPA = \binom{N}{2}^{-1} \sum_{i=0}^{M_t} \sum_{j=i+1}^{M_t} 1 - |p^g_{ij} - p^e_{ij}| \quad (2)$$

Thus, SPA rewards an evaluator for expressing confidence levels similar to those of the ground truth and penalizes deviations.

C.2 Pairwise Accuracy with Tie Calibration (acc^*_{eq})

Deutsch et al. (2023) introduced acc^*_{eq} to account for tied scores in meta-evaluation. Unlike PA and SPA, acc^*_{eq} is a segment-level measure, meaning it evaluates a metric's ability to estimate the quality of individual translations rather than MT systems. Specifically, acc^*_{eq} counts how often an evaluator e ranks pairs of translations of the same source segment in the same order as the ground truth g, accounting for tied scores.

Let C be the number of translation pairs ranked in the same order by both the evaluator e and the ground truth g. Similarly, let D denote the pairs ranked in the opposite order. The terms T_e and T_g represent pairs tied only in the evaluator's scores and only in the ground truth, respectively. Lastly, T_{eg} refers to pairs tied in both the evaluator's scores and the ground truth. acc^*_{eq} is then defined as:

$$\text{acc}^*_{eq} = \frac{C + T_{eg}}{C + D + T_e + T_g + T_{eg}} \quad (3)$$

Tie Calibration Many automatic metrics produce assessments on a continuous scale, such as the real numbers in the interval $[0, 1]$. As a consequence, these metrics rarely, if ever, produce tied scores, resulting in $T_e \approx 0$ and $T_{eg} \approx 0$. The Tie

Calibration algorithm addresses this issue by estimating a threshold value ϵ_e for each evaluator e, such that two assessments e_i and e_j are considered tied if $|e_i - e_j| \leq \epsilon_e$.

D Metrics

Table 3 lists the complete set of automatic evaluators considered in this work.

E Full Rankings

Tables 4, 5, 6, and 7 present the same rankings of Table 2, but including all tested evaluators.

F Full Rankings Varying the Ground Truth

In this section, we examine how evaluator rankings vary depending on the choice of human evaluator used as ground truth. Specifically, we use the following evaluators as ground truth:

- pSQM-1 from Table 4.
- DA+SQM from Table 6.
- MQM-2023-2 from Table 6.
- MQM-2023-3 from Table 6.
- ESA from Table 7.

We exclude the ESA-1, ESA-2, and MQM-2023-4 evaluators from the 2023 EN→DE test set, as they annotated only a limited number of segments. This increases the number of segments in the 2023 EN→DE test set from 145 to 376. Therefore, for reference, we also report results on this test set using the same evaluator as in Tables 2 and 6. Results are presented in Tables 8, 9, 10, 11, 12, and 13.

As we can see, our findings remain valid when varying the evaluator selected as ground truth, with human evaluators consistently ranking the same as or lower than automatic metrics.

Metric	Reference paper	Metric	Reference paper
all-rembert-20	(Mathur et al., 2020)	metametrics_mt_mqm	(Anugraha et al., 2024)
BAQ_dyn	(Mathur et al., 2020)	metametrics_mt_mqm_qe	(Anugraha et al., 2024)
BAQ_static	(Mathur et al., 2020)	MetricX-23-QE-XXL	(Juraska et al., 2023)
BERT-base-L2	(Mathur et al., 2020)	MetricX-23-XXL	(Juraska et al., 2023)
BERT-large-L2	(Mathur et al., 2020)	MetricX-24-Hybrid	(Juraska et al., 2024)
BERTScore	(Zhang et al., 2020)	MetricX-24-Hybrid-QE	(Juraska et al., 2024)
BLCOM_1	(Freitag et al., 2024)	metricx_xxl_MQM_2020	(Freitag et al., 2022)
BLEU	(Papineni et al., 2002)	mre-score-labse-regular	(Viskov et al., 2023)
BLEURT	(Sellam et al., 2020a)	MS-COMET-22	(Kocmi et al., 2022b)
BLEURT-0.1-all	(Mathur et al., 2020)	MS-COMET-QE-22	(Kocmi et al., 2022b)
BLEURT-0.1-en	(Mathur et al., 2020)	OpenKiwi-Bert	(Kepler et al., 2019)
BLEURT-0.2	(Mathur et al., 2020)	OpenKiwi-XLMR	(Kepler et al., 2019)
BLEURT-20	(Sellam et al., 2020a)	parbleu	(Bawden et al., 2020)
bleurt-combi	(Mathur et al., 2020)	parchrf++	(Bawden et al., 2020)
BLEURT-extended	(Sellam et al., 2020b)	paresim-1	(Bawden et al., 2020)
bright-qe	(Freitag et al., 2024)	prism	(Thompson and Post, 2020a)
Calibri-COMET22	(Freitag et al., 2023)	prismRef	(Thompson and Post, 2020a,b)
Calibri-COMET22-QE	(Freitag et al., 2023)	PrismRefMedium	(Thompson and Post, 2020a,b)
CharacTER	(Wang et al., 2016)	PrismRefSmall	(Thompson and Post, 2020a,b)
chrF	(Popović, 2015)	prismSrc	(Thompson and Post, 2020a,b)
chrF++	(Popović, 2017)	Random-sysname	(Freitag et al., 2023)
chrfS	(Mukherjee and Shrivastava, 2024)	REUSE	(Mukherjee and Shrivastava, 2022a)
COMET	(Rei et al., 2020b)	sentBLEU	(Papineni et al., 2002)
COMET-20	(Rei et al., 2020a)	sentinel-cand-mqm	(Perrella et al., 2024b)
COMET-22	(Rei et al., 2022a)	sentinel-ref-mqm	(Perrella et al., 2024b)
COMET-2R	(Rei et al., 2020b)	sentinel-src-mqm	(Perrella et al., 2024b)
COMET-HTER	(Rei et al., 2020b)	SEScore	(Xu et al., 2022)
COMET-MQM	(Rei et al., 2020b)	sescoreX	(Xu et al., 2023)
COMET-QE	(Rei et al., 2021)	spBLEU	(Team et al., 2022)
COMET-Rank	(Rei et al., 2020b)	SWSS+METEOR	(Xu et al., 2020)
COMETKiwi	(Rei et al., 2022b)	TER	(Snover et al., 2006)
CometKiwi-XL	(Rei et al., 2023)	tokengram_F	(Dreano et al., 2023b)
CometKiwi-XXL	(Rei et al., 2023)	UniTE	(Wan et al., 2022b,a)
cometoid22-wmt22	(Gowda et al., 2023)	UniTE-src	(Wan et al., 2022b)
damonmonli	(Freitag et al., 2024)	XCOMET	(Guerreiro et al., 2024)
docWMT22CometDA	(Vernikos et al., 2022)	XCOMET-Ensemble	(Guerreiro et al., 2024)
docWMT22CometKiwiDA	(Vernikos et al., 2022)	XCOMET-QE	(Guerreiro et al., 2024)
eBLEU	(ElNokrashy and Kocmi, 2023)	XCOMET-QE-Ensemble	(Guerreiro et al., 2024)
EED	(Stanchev et al., 2019)	XLsim	(Mukherjee and Shrivastava, 2023)
embed_llama	(Dreano et al., 2023a)	XLsimMqm	(Mukherjee and Shrivastava, 2023)
esim	(Mathur et al., 2019)	YiSi-0	(Lo, 2019)
f200spBLEU	(Team et al., 2022)	YiSi-1	(Lo, 2019)
GEMBA-MQM	(Kocmi and Federmann, 2023a)	YiSi-2	(Lo, 2019)
gemba_esa	(Freitag et al., 2024)	Yisi-combi	(Mathur et al., 2020)
HWTSC-Teacher-Sim	(Liu et al., 2022)	yisi1-translate	(Mathur et al., 2020)
KG-BERTScore	(Wu et al., 2023)	mbr-metricx-qe	(Naskar et al., 2023)
MaTESe	(Perrella et al., 2022)	mBERT-L2	(Mathur et al., 2020)
MaTESe-QE	(Perrella et al., 2022)	MEE	(Mukherjee et al., 2020)
MEE4	(Mukherjee and Shrivastava, 2022b)		

Table 3: List of all automatic evaluators considered, i.e., MT metrics, associated with their reference papers. Metrics without dedicated papers cite the Metrics Shared Task results paper in which they appeared.

	EN→DE				ZH→EN			
	SPA		acc^*_{eq}		SPA		acc^*_{eq}	
Metric	Rank	Acc.	Rank	Acc.	Rank	Acc.	Rank	Acc.
MQM-2020-2	1	96.45	1	58.86	1	88.10	1	55.70
pSQM-1	1	95.59	6	49.41	1	79.16	13	43.89
MQM-2020-3	2	90.39	2	56.84	1	92.06	2	52.80
BLEURT-0.2	2	86.81	4	50.81	2	72.59	3	50.57
pSQM-2	2	85.87	9	46.97	1	89.33	9	46.77
BLEURT-20	2	85.52	3	51.68	3	67.46	4	50.12
pSQM-3	2	84.61	6	49.38	1	87.94	7	47.88
all-rembert-20	3	79.19	4	51.04	3	66.41	3	50.61
BLEURT-extended	3	75.55	5	50.21	3	64.00	3	50.74
COMET-MQM	4	71.39	7	48.21	4	55.43	6	48.49
BLEURT-0.1-all	4	71.38	7	48.63	2	71.04	5	49.54
COMET	4	71.09	8	47.36	4	56.01	5	49.28
COMET-QE*	4	70.59	8	47.82	3	58.37	8	47.09
COMET-HTER	5	65.71	8	47.62	4	54.79	5	49.30
mBERT-L2	5	65.03	10	45.48	4	56.49	6	48.97
COMET-2R	6	58.12	9	46.43	4	55.99	4	50.20
COMET-Rank	6	54.78	14	41.31	3	58.16	14	43.57
OpenKiwi-XLMR*	6	53.25	11	44.11	4	53.29	8	47.23
OpenKiwi-Bert*	6	52.01	16	39.98	3	59.55	11	45.13
prism	6	51.92	11	43.59	4	57.88	8	47.56
Yisi-combi	7	51.10	12	42.63	–	–	–	–
bleurt-combi	7	51.10	12	42.63	–	–	–	–
esim	7	50.72	14	41.35	4	52.90	10	46.19
chrF	7	49.86	13	42.05	5	47.70	13	44.09
EED	7	49.81	15	40.94	5	45.41	14	43.64
paresim-1	7	49.54	14	41.37	4	53.34	10	46.15
chrF++	7	48.87	13	41.99	5	48.96	12	44.27
YiSi-1	7	48.79	12	42.70	4	52.74	7	48.01
CharacTER	7	47.71	16	40.45	5	48.84	13	44.01
BLEURT-0.1-en	7	47.43	15	40.96	4	57.29	7	48.26
YiSi-0	7	46.23	17	39.78	5	46.47	14	43.60
TER	7	45.98	16	40.15	6	39.68	15	43.34
parchrf++	7	45.57	13	42.25	5	48.68	12	44.25
MEE	7	45.31	14	41.61	4	52.91	13	43.94
sentBLEU	7	44.41	15	41.07	4	50.45	15	43.37
parbleu	8	41.38	15	41.01	4	50.28	15	43.43
yisi1-translate	8	39.76	12	42.60	4	52.28	11	44.70
YiSi-2*	8	38.44	18	34.36	5	43.35	12	44.60

Table 4: 2020

	EN→DE				EN→ZH			
	SPA		acc^*_{eq}		SPA		acc^*_{eq}	
Metric	Rank	Acc.	Rank	Acc.	Rank	Acc.	Rank	Acc.
MetricX-23-QE-XXL*	1	94.89	3	57.64	2	83.92	2	47.43
MQM-2022-2	1	94.49	6	55.55	2	80.82	3	47.05
MQM-2022-3	1	92.59	1	61.06	1	87.22	2	47.56
MetricX-23-XXL	2	92.34	2	59.27	1	87.69	1	48.43
COMET-22	2	91.63	5	56.51	2	84.08	3	46.74
COMET-20	2	91.28	9	52.42	2	80.56	7	43.81
CometKiwi*	2	89.51	7	53.77	3	75.36	8	43.21
BLEURT-20	3	88.20	7	53.33	3	77.80	7	43.84
metricx_xxl_MQM_2020	3	88.10	3	57.43	1	87.04	3	46.89
COMET-QE*	3	85.51	10	51.69	3	78.33	7	43.61
MS-COMET-22	3	85.37	8	53.13	1	85.18	6	44.92
CometKiwi-XXL*	3	84.43	7	53.27	2	81.25	2	47.28
UniTE	4	82.77	4	57.03	2	83.88	5	45.86
UniTE-src*	4	81.55	6	55.00	4	65.74	7	43.53
CometKiwi-XL*	4	81.13	8	52.73	2	81.56	4	46.33
YiSi-1	4	78.91	13	48.26	4	70.72	8	43.23
MATESE	5	78.03	7	53.48	–	–	–	–
BERTScore	5	75.61	14	47.57	4	70.69	8	43.28
SEScore	5	75.16	12	50.45	–	–	–	–
MS-COMET-QE-22*	5	74.44	12	50.37	2	78.84	9	42.51
MEE4	5	74.19	15	46.81	–	–	–	–
chrF	5	73.05	16	46.38	3	72.67	10	41.87
f200spBLEU	5	71.04	15	46.84	4	71.76	10	41.85
HWTSC-Teacher-Sim*	5	69.68	13	48.10	4	68.43	11	40.53
DA+SQM	6	66.61	16	46.03	2	82.95	12	36.26
MATESE-QE*	6	65.42	11	51.06	–	–	–	–
BLEU	6	65.00	15	46.51	4	67.31	13	34.28
REUSE*	7	37.95	17	43.58	5	33.46	12	35.89

Table 5: 2022

	EN→DE				ZH→EN			
	SPA		acc^*_{eq}		SPA		acc^*_{eq}	
Metric	Rank	Acc.	Rank	Acc.	Rank	Acc.	Rank	Acc.
GEMBA-MQM*	1	94.52	5	58.52	1	93.17	3	52.80
MQM-2023-3	1	93.51	5	58.42	1	95.54	5	51.65
CometKiwi-XXL*	1	93.22	5	58.46	1	92.86	6	50.94
MQM-2023-2	1	93.15	6	57.71	1	95.18	2	52.90
CometKiwi-XL*	1	93.11	6	57.38	2	92.02	6	50.71
MetricX-23-XXL	1	92.57	2	61.82	2	91.58	2	53.13
XCOMET-QE-Ensemble*	1	92.48	4	59.89	2	90.54	3	52.87
cometoid22-wmt22*	1	92.43	5	58.09	2	90.09	7	50.23
COMET	1	92.33	7	56.65	4	87.18	9	48.42
XCOMET-Ensemble	1	92.21	3	60.99	2	91.15	1	54.59
MetricX-23-QE-XXL*	1	92.12	1	62.53	3	88.30	2	53.26
Calibri-COMET22	1	92.01	10	51.26	4	87.02	16	44.57
docWMT22CometDA	1	91.76	8	54.71	4	87.41	13	46.15
sescoreX	1	91.66	8	54.76	4	85.73	13	46.39
DA+SQM	2	91.24	14	46.79	4	86.28	22	39.42
Calibri-COMET22-QE*	2	90.80	11	50.33	4	87.59	11	47.24
ESA-1	2	90.39	14	46.71	–	–	–	–
BLEURT-20	2	90.35	8	55.19	4	87.36	9	48.63
mbr-metricx-qe*	2	89.98	5	58.75	3	88.55	4	52.04
prismRef	2	89.92	11	50.72	5	82.50	14	46.06
docWMT22CometKiwiDA*	2	89.92	8	55.30	2	90.95	10	47.83
MS-COMET-QE-22*	2	89.85	8	54.55	4	87.59	10	47.81
f200spBLEU	2	89.24	11	50.54	5	81.12	18	43.33
CometKiwi*	2	89.23	6	57.74	3	89.37	5	51.74
mre-score-labse-regular	2	89.14	10	51.12	4	87.14	17	43.80
ESA-2	2	89.11	12	49.70	–	–	–	–
YiSi-1	2	88.96	9	53.15	4	85.70	12	46.68
MQM-2023-4	2	88.93	14	46.68	–	–	–	–
KG-BERTScore*	2	88.79	7	56.98	3	89.31	8	49.75
MaTESe	2	88.40	9	53.36	2	92.06	7	50.34
BLEU	2	88.02	12	50.06	6	80.92	19	43.13
BERTscore	2	87.33	11	50.88	5	84.68	15	45.79
MEE4	2	87.07	10	51.62	6	80.51	19	42.94
XLsim	2	86.58	10	51.01	6	81.00	19	42.84
tokengram_F	3	85.60	12	49.72	5	81.01	18	43.52
chrF	4	84.25	12	49.54	5	81.47	17	43.72
eBLEU	4	83.87	13	48.96	6	80.44	20	42.55
embed_llama	4	81.33	14	47.12	4	84.84	21	41.05
Random-sysname*	5	59.47	16	39.07	7	54.34	23	34.49
prismSrc*	6	30.03	15	40.89	8	35.54	22	39.28

Table 6: 2023

	EN→ES			
	SPA		acc^*_{eq}	
Metric	Rank	Acc.	Rank	Acc.
CometKiwi-XXL*	1	86.12	4	67.24
gemba_esa*	1	85.72	3	67.68
COMET-22	1	82.37	5	66.60
bright-qe*	1	81.77	4	67.39
ESA	2	80.12	8	63.84
XCOMET-QE*	2	80.10	3	67.99
metametrics_mt_mqm_hybrid_kendall	2	80.10	1	68.95
XCOMET	2	79.96	2	68.67
MetricX-24-Hybrid	2	79.75	1	69.20
BLCOM_1	2	79.17	6	65.02
MetricX-24-Hybrid-QE*	2	79.09	2	68.92
sentinel-cand-mqm*	2	78.54	5	66.39
BLEURT-20	2	75.96	7	64.48
metametrics_mt_mqm_qe_kendall.seg.s*	3	73.29	4	67.49
CometKiwi*	3	71.74	5	66.51
PrismRefMedium	3	70.93	11	61.39
PrismRefSmall	3	70.52	10	61.51
YiSi-1	3	70.51	11	61.44
BERTScore	3	67.75	11	61.41
chrF	3	66.73	13	61.05
damonmonli	3	66.37	9	62.10
chrfS	4	64.31	11	61.37
spBLEU	4	63.19	12	61.08
BLEU	5	60.67	13	61.04
MEE4	5	60.36	10	61.57
sentinel-ref-mqm	6	44.19	13	61.04
sentinel-src-mqm*	6	44.19	13	61.04
XLsimMqm*	6	39.25	12	61.11

Table 7: 2024

	ZH→EN			
	SPA		acc^*_{eq}	
Metric	Rank	Acc.	Rank	Acc.
pSQM-3	1	83.42	6	65.23
MQM-2020-2	1	80.56	6	65.22
MQM-2020-3	1	80.34	6	65.22
MQM-2020-1	1	79.55	6	65.22
pSQM-2	1	74.02	4	65.26
BERT-large-L2	1	70.19	3	65.38
COMET	1	68.58	1	65.64
SWSS+METEOR	2	67.26	4	65.26
MEE	2	67.07	6	65.22
prism	2	66.01	5	65.25
sentBLEU	2	65.70	5	65.25
parbleu	2	65.63	6	65.23
BLEURT	2	65.28	2	65.49
YiSi-1	2	64.71	6	65.22
yisi1-translate	2	64.32	6	65.23
CharacTER	2	63.99	6	65.23
all-rembert-20	2	63.48	2	65.47
BLEURT-20	2	63.10	3	65.38
paresim-1	2	62.79	4	65.30
esim	2	62.42	4	65.30
chrF++	3	62.38	6	65.23
BLEURT-0.1-en	3	62.22	2	65.47
COMET-2R	3	62.10	1	65.61
parchrf++	3	61.92	5	65.24
EED	3	60.91	6	65.23
mBERT-L2	3	60.84	2	65.43
chrF	3	60.81	4	65.25
YiSi-0	3	60.55	5	65.25
BLEURT-0.2	3	60.54	3	65.34
COMET-Rank	3	59.97	6	65.22
BLEURT-extended	3	59.90	3	65.36
BAQ_static	3	59.89	6	65.22
BLEURT-0.1-all	3	59.80	4	65.25
BERT-base-L2	3	59.57	2	65.49
COMET-HTER	3	58.88	2	65.42
COMET-MQM	3	58.27	5	65.25
BAQ_dyn	3	58.12	6	65.22
COMET-QE*	3	57.83	4	65.26
TER	3	56.12	5	65.25
OpenKiwi-Bert*	3	55.35	5	65.25
OpenKiwi-XLMR*	3	50.98	4	65.30
YiSi-2*	4	48.35	4	65.27

Table 8: The test set is 2020 ZH→EN. The evaluator selected as the ground truth follows the pSQM protocol (pSQM-1 in Table 4).

	EN→DE			
	SPA		acc^*_{eq}	
Metric	Rank	Acc.	Rank	Acc.
MetricX-23-QE-XXL*	1	95.11	3	58.41
GEMBA-MQM*	1	94.89	14	43.06
CometKiwi*	1	94.83	3	58.21
KG-BERTScore*	1	94.57	5	57.16
MQM-2023-3	1	94.24	13	46.99
docWMT22CometKiwiDA*	1	94.22	2	58.60
CometKiwi-XL*	1	94.16	2	58.80
MS-COMET-QE-22*	1	94.04	7	55.66
CometKiwi-XXL*	1	93.98	1	59.66
MetricX-23-XXL	2	93.31	3	58.06
COMET	2	92.80	3	58.42
docWMT22CometDA	2	92.45	2	58.95
mre-score-labse-regular	2	92.30	8	54.75
MQM-2023-1	2	92.10	15	42.20
Calibri-COMET22	2	92.02	3	58.17
mbr-metricx-qe*	2	91.94	3	58.26
MQM-2023-2	2	91.40	11	48.91
cometoid22-wmt22*	2	91.34	5	57.06
sescoreX	2	91.22	4	57.41
BLEURT-20	3	90.66	5	57.26
prismRef	3	89.58	10	54.16
Calibri-COMET22-QE*	3	89.55	8	55.22
YiSi-1	3	89.41	6	56.64
XLsim	3	87.93	6	56.47
XCOMET-Ensemble	4	87.77	4	57.82
XCOMET-QE-Ensemble*	4	87.34	6	56.40
eBLEU	4	87.33	10	53.91
BERTscore	4	86.83	7	56.13
f200spBLEU	4	86.82	8	54.97
MaTESe	4	86.69	16	37.35
MEE4	4	86.12	7	55.93
BLEU	5	84.45	10	53.63
tokengram_F	5	83.15	8	54.98
chrF	5	82.45	9	54.74
embed_llama	5	81.27	9	54.28
Random-sysname*	6	60.55	12	47.94
prismSrc*	7	28.59	11	48.54

Table 9: The test set is 2023 EN→DE. The evaluator selected as the ground truth follows the DA+SQM protocol (DA+SQM in Table 6). Different from Tables 2 and 6, we exclude the evaluators ESA-1, ESA-2, and MQM-2023-4, because they annotated a limited number of translations. This way, we increase the number of segments in the test set from 145 to 376.

	EN→DE			
	SPA		acc^*_{eq}	
Metric	Rank	Acc.	Rank	Acc.
MQM-2023-3	1	97.09	7	56.62
GEMBA-MQM*	1	97.09	5	59.18
CometKiwi-XXL*	1	95.96	7	56.43
CometKiwi-XL*	1	95.47	6	57.33
MQM-2023-2	1	95.20	4	60.04
docWMT22CometDA	1	94.99	10	53.86
MetricX-23-XXL	2	94.93	2	61.65
XCOMET-Ensemble	2	94.58	1	62.23
MetricX-23-QE-XXL*	2	94.52	3	61.14
COMET	2	94.39	8	55.71
XCOMET-QE-Ensemble*	2	94.14	4	59.89
docWMT22CometKiwiDA*	2	93.70	10	53.55
BLEURT-20	2	93.35	9	54.83
Calibri-COMET22-QE*	2	93.25	14	49.62
cometoid22-wmt22*	2	92.43	7	56.73
CometKiwi*	3	92.27	7	56.76
DA+SQM	3	92.10	18	43.68
KG-BERTScore*	3	92.05	9	54.40
sescoreX	3	91.99	10	53.99
YiSi-1	3	91.28	11	51.66
mbr-metricx-qe*	3	91.22	8	56.19
MS-COMET-QE-22*	3	90.70	10	53.51
prismRef	3	90.22	14	49.74
Calibri-COMET22	3	88.28	14	49.54
XLsim	4	88.01	13	50.31
mre-score-labse-regular	4	87.47	13	49.85
BERTscore	4	87.01	12	50.48
f200spBLEU	4	86.90	13	50.37
MaTESe	4	86.73	7	56.23
eBLEU	4	86.24	16	48.30
MEE4	4	86.05	12	50.81
BLEU	5	84.48	15	49.18
tokengram_F	5	83.85	14	49.51
chrF	5	83.34	14	49.44
embed_llama	5	80.09	17	45.05
Random-sysname*	6	61.26	20	38.19
prismSrc*	7	28.88	19	40.02

Table 10: The test set is 2023 EN→DE. The evaluator selected as the ground truth follows the MQM protocol (it is the evaluator selected as ground truth in Table 6). Different from Tables 2 and 6, we exclude the evaluators ESA-1, ESA-2, and MQM-2023-4, because they annotated a limited number of translations. This way, we increase the number of segments in the test set from 145 to 376.

	EN→DE			
	SPA		acc^*_{eq}	
Metric	Rank	Acc.	Rank	Acc.
BLEURT-20	1	97.12	6	60.66
MetricX-23-XXL	1	96.51	1	64.61
docWMT22CometDA	1	96.41	8	59.16
CometKiwi-XXL*	1	96.40	4	61.57
GEMBA-MQM*	1	96.19	8	58.96
CometKiwi-XL*	1	95.88	6	60.45
COMET	1	95.82	5	61.31
MQM-2023-3	1	95.50	8	59.25
MQM-2023-1	1	95.20	7	60.03
mbr-metricx-qe*	1	95.12	3	62.39
MetricX-23-QE-XXL*	2	94.99	2	63.51
Calibri-COMET22-QE*	2	94.48	15	51.95
docWMT22CometKiwiDA*	2	94.47	9	57.58
XCOMET-Ensemble	2	93.81	1	64.20
sescoreX	2	93.51	7	59.94
CometKiwi*	2	93.04	6	60.24
KG-BERTScore*	2	92.78	8	59.06
XCOMET-QE-Ensemble*	3	92.71	4	61.92
DA+SQM	3	91.40	16	48.92
MS-COMET-QE-22*	3	91.34	9	57.54
cometoid22-wmt22*	3	91.22	5	61.07
YiSi-1	3	90.47	9	57.50
mre-score-labse-regular	3	90.02	10	56.50
prismRef	3	89.92	12	55.16
f200spBLEU	3	89.09	11	55.81
XLsim	4	88.77	11	55.49
Calibri-COMET22	4	88.37	11	55.67
eBLEU	4	88.19	14	54.13
BERTscore	4	88.09	10	56.27
BLEU	4	87.09	13	54.61
MaTESe	4	86.49	13	54.19
MEE4	4	86.46	10	56.41
tokengram_F	4	86.12	11	55.66
chrF	4	85.53	11	55.50
embed_llama	5	80.95	15	51.39
Random-sysname*	6	59.64	18	42.43
prismSrc*	7	25.26	17	43.73

Table 11: The test set is 2023 EN→DE. The evaluator selected as the ground truth follows the MQM protocol (MQM-2023-2 in Table 6). Different from Tables 2 and 6, we exclude the evaluators ESA-1, ESA-2, and MQM-2023-4, because they annotated a limited number of translations. This way, we increase the number of segments in the test set from 145 to 376.

	EN→DE			
	SPA		acc^*_{eq}	
Metric	Rank	Acc.	Rank	Acc.
GEMBA-MQM*	1	97.98	7	55.71
CometKiwi-XXL*	1	97.96	4	58.36
CometKiwi-XL*	1	97.49	5	57.78
MQM-2023-1	1	97.09	6	56.46
MetricX-23-XXL	1	96.86	1	61.26
docWMT22CometDA	1	96.82	6	56.98
MetricX-23-QE-XXL*	1	96.47	2	60.55
COMET	2	96.30	4	58.63
docWMT22CometKiwiDA*	2	96.09	8	54.86
MQM-2023-2	2	95.50	3	59.25
CometKiwi*	2	94.67	5	57.83
KG-BERTScore*	2	94.45	6	56.67
sescoreX	2	94.31	5	57.58
DA+SQM	2	94.24	14	46.99
BLEURT-20	3	94.18	4	58.68
Calibri-COMET22-QE*	3	94.01	13	49.98
YiSi-1	3	93.19	7	55.64
MS-COMET-QE-22*	3	93.17	7	55.46
mbr-metricx-qe*	3	93.06	3	59.30
XCOMET-Ensemble	3	92.64	2	60.61
prismRef	3	92.58	10	53.57
cometoid22-wmt22*	3	92.55	4	58.50
XCOMET-QE-Ensemble*	3	92.14	3	59.43
Calibri-COMET22	4	90.86	10	53.18
XLsim	4	90.86	8	54.63
BERTscore	4	89.67	9	54.30
f200spBLEU	4	89.56	9	54.06
mre-score-labse-regular	4	89.14	8	54.47
MEE4	4	88.72	8	54.40
eBLEU	4	88.67	11	53.06
BLEU	5	87.13	11	52.76
tokengram_F	5	86.17	10	53.62
chrF	5	85.58	10	53.51
MaTESe	5	84.75	12	52.12
embed_llama	5	82.93	13	50.10
Random-sysname*	6	58.48	16	41.27
prismSrc*	7	28.67	15	44.52

Table 12: The test set is 2023 EN→DE. The evaluator selected as the ground truth follows the MQM protocol (MQM-2023-3 in Table 6). Different from Tables 2 and 6, we exclude the evaluators ESA-1, ESA-2, and MQM-2023-4, because they annotated a limited number of translations. This way, we increase the number of segments in the test set from 145 to 376.

	EN→ES			
	SPA		acc^*_{eq}	
Metric	Rank	Acc.	Rank	Acc.
COMET-22	1	86.90	2	53.11
BLCOM_1	1	86.12	2	53.00
XCOMET	1	84.88	3	52.35
metametrics_mt_mqm_hybrid_kendall	1	84.80	1	53.92
XCOMET-QE*	1	83.67	4	51.01
PrismRefMedium	1	83.07	5	50.62
MetricX-24-Hybrid	2	82.96	2	53.13
BLEURT-20	2	82.19	2	52.93
gemba_esa*	2	81.37	10	40.86
MetricX-24-Hybrid-QE*	2	80.93	4	51.35
PrismRefSmall	2	80.84	4	51.23
MQM-2024	2	80.13	11	34.61
sentinel-cand-mqm*	2	79.06	6	50.23
YiSi-1	2	78.76	3	51.88
BERTScore	2	78.28	4	50.88
metametrics_mt_mqm_qe_kendall.seg.s*	3	77.05	7	49.17
CometKiwi-XXL*	3	76.51	5	50.86
bright-qe*	3	75.73	9	43.46
MEE4	3	75.20	4	51.03
CometKiwi*	3	74.39	5	50.47
chrfS	3	73.90	4	51.05
chrF	4	70.94	5	50.69
spBLEU	4	70.77	6	49.81
BLEU	4	69.37	7	49.43
damonmonli	4	63.34	8	48.14
sentinel-ref-mqm	5	54.36	12	15.38
sentinel-src-mqm*	5	54.36	12	15.38
XLsimMqm*	5	39.32	9	43.02

Table 13: The test set is 2024. The evaluator selected as the ground truth follows the ESA protocol (ESA in Table 7).

Diffusion Directed Acyclic Transformer for Non-Autoregressive Machine Translation

Quan Nguyen-Tri[1], Cong Dao Tran[1], Hoang Thanh-Tung[2],
[1]FPT Software AI Center,
[2]University of Engineering and Technology, Vietnam National University Hanoi,
quannt40@fpt.com, daotc2@fpt.com, hoangthanhtung53ca@vnu.edu.vn

Abstract

Non-autoregressive transformers (NATs) predict entire sequences in parallel to reduce decoding latency, but they often encounter performance challenges due to the multi-modality problem. A recent advancement, the Directed Acyclic Transformer (DAT), addresses this issue by capturing multiple translation modalities to paths in a Directed Acyclic Graph (DAG). However, the collaboration with the latent variable introduced through the Glancing training (GLAT) is crucial for DAT to attain state-of-the-art performance. In this paper, we introduce Diffusion Directed Acyclic Transformer (Diff-DAT), which serves as an alternative to GLAT as a latent variable introduction for DAT. Diff-DAT offers two significant benefits over the previous approach. Firstly, it establishes a stronger alignment between training and inference. Secondly, it facilitates a more flexible tradeoff between quality and latency.

1 Introduction

The Transformer architecture (Vaswani et al., 2017a) has gained immense popularity, particularly in sequence-to-sequence learning problems like machine translation. Conventional Transformers employ an autoregressive approach to generation, yielding robust results but proving inefficient at inference due to its sequential decoding. To address this issue, Non-autoregressive Transformers (NATs) (Gu et al., 2018) was introduced, significantly boosting the decoding speed by generating all output tokens simultaneously. This advantage often comes with a trade-off in translation quality due to the challenging multi-modality problem (Gu et al., 2018), wherein a single source sentence may have multiple translations in the target language.

Numerous approaches have been proposed to address this challenge, primarily by introducing additional latent variables to reduce the number of translation modalities given the latent variables. Among them, the Directed Acyclic Transformer (DAT) (Huang et al., 2022b) emerges as the most promising approach. In DAT, translation modalities are assigned to paths in a Directed Acyclic Graph (DAG), enabling the model to capture multiple translation modalities. Although DAT enhances translation quality and diversity, it still requires additional context from the target as a latent variable to perform effectively. Huang et al. (2022b) demonstrated that the latent variable from Glancing training (GLAT) (Qian et al., 2021a) significantly improves DAT's performance.

In this work, we aimed to enhance the capabilities of DAT by introducing latent variables through a diffusion process. We integrated diffusion models into DAT, resulting in a novel model called Diffusion Directed Acyclic Transformer (Diff-DAT). We discovered that the diffusion model can effectively replace GLAT as a latent variable introduction mechanism, enabling DAT to function optimally. This integration offers two significant advantages. Firstly, diffusion models establish a stronger alignment between training and inference. Secondly, they facilitate a more flexible tradeoff between quality and speed by allowing decoding through multiple iterations. Results on multiple machine translation benchmarks demonstrate that our approach not only improves the performance of DAT on fully non-autoregressive decoding but also improves its iterative decoding performance without a significant increase in decoding latency.

2 Preliminaries

DA-Transformer replaces the traditional Transformer's decoder with a directed acyclic decoder. This decoder organizes its outputs as a directed acyclic graph (DAG), where each path corresponds to a specific translation modality. Given a bilingual pair, $X = \{x^1, \ldots, x^N\}$ and $Y = \{y^1, \ldots, y^M\}$, DAT sets the decoder length to $L = \lambda \cdot N$ and

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 814–828
July 27 - August 1, 2025

models the translation probability by marginalizing out the paths in the DAG.

$$P_\theta(Y|X) = \sum_A P_\theta(Y|A,X)P_\theta(A|X), \quad (1)$$

where $A = \{a^1, ..., a^M\}$ represents a path with vertex indexes satisfying $1 = a^1 < ... < a^M = L$. $P_\theta(A|X)$ and $P_\theta(Y|A,X)$ denote the probability of path A and the probability of target sentence Y conditioned on path A, respectively. The DAG factorizes the path probability $P_\theta(A|X)$ based on the Markov assumption:

$$P_\theta(A|X) = \prod_{i=1}^{M-1} P_\theta(a^{i+1}|a^i, X) = \prod_{i=1}^{M-1} \mathbf{E}_{a^i,a^{i+1}}, \quad (2)$$

where $\mathbf{E} \in \mathbb{R}^{L\times L}$ is a row-normalized transition matrix. The DAG's unidirectional nature masks the lower triangular part of $\mathbf{E}$ to zeros. Once the path A is determined, token y^i is generated conditioned on the decoder hidden state with index a^i:

$$P_\theta(Y|A,X) = \prod_{i=1}^{M} P_\theta(y^i|a^i, X), \quad (3)$$

where $P_\theta(y^i|a^i, X)$ represents the translation probability of word y^i on the position a^i of decoder. To enhance the performance of DAT, GLAT was incorporated as an additional latent variable, denoted as Z. This variable is a randomly masked target and serves as an extra input for the decoder. The final training objective is to maximize the log-likelihood with the additional latent variable:

$$\log P_\theta(Y|X) = \mathbb{E}_{Q(Z|Y,\hat{A})} \log P_\theta(Y|X,Z), \quad (4)$$

since the decoder input is longer than the target sentence, (Huang et al., 2022b) first finds the most probable path $\hat{A} = \arg\max_A P_\theta(Y, A|X)$ and uses it to assign the masked target to vertices in the DAG.

3 Methodology

Training objective: The latent variable Z in GLAT is only used during training when a target is provided and is omitted during inference, forcing the model to rely solely on X to predict the target. This mismatch between training and inference can damage the model's generalizability. This motivates us to incorporate diffusion models into DAT.

Diffusion models (Sohl-Dickstein et al., 2015) aim to predict the target Y_0 through a sequence of latent variables $Y_{1:T} = Y_1, Y_2, \ldots, Y_T$. The forward process gradually adds noise to the target Y_0 over T steps to get Y_T, the final latent variable that follows a prior noise distribution. The backward process optimizes a neural network to denoise the noisy latent variables, reversing the forward process to recover Y_0. When the step size is infinitesimally small, the forward and backward processes have the same functional form.

While diffusion models provide a strong theoretical justification for aligning training and inference, the large number of time steps (T) hinders their practical application in NATs, where decoding latency is a critical concern. Therefore, we utilize absorbing discrete diffusion (Austin et al., 2021) that uses the absorbing state ($[M]$) as noise to add to the target sentence at each step, until all tokens are masked (noise distribution Y_T). This approach reduces the number of forward steps T required to reach the noise distribution.

We compute the translation probability by marginalizing out the paths in the DAG and latent variables from the diffusion backward process.

$$\begin{aligned} P_\theta(Y_0|X) &= \sum_{Y_{1:T}} \sum_A P_\theta(Y_{0:T}, A|X) \\ &= \sum_{Y_{1:T}} \sum_A P(Y_T) \prod_{t=1}^{T} P_\theta(Y_{t-1}, A|Y_t, X) \end{aligned} \quad (5)$$

Diff-DAT maximizes the variational lower bound (VLB) of the log-likelihood:

$$\begin{aligned} &\log P_\theta(Y_0|X) = \log \sum_{Y_{1:T}} \sum_A P_\theta(Y_{0:T}, A|X) \\ &\geq \sum_A \mathbb{E}_{Q(Y_{1:T}|Y_0,A)} \log \frac{P_\theta(Y_{0:T}, A|X)}{Q(Y_{1:T}|Y_0, A)} \\ &\approx \mathbb{E}_{Q(Y_{1:T}|Y_0,\hat{A})} \sum_A \log \frac{P_\theta(Y_{0:T}, A|X)}{Q(Y_{1:T}|Y_0, \hat{A})} \\ &= \sum_{t=1}^{T} \mathcal{L}_t + \text{const}, \end{aligned} \quad (6)$$

Where $Q(Y_{1:T}|Y_0, \hat{A})$ represents the forward process transition probabilities. Following Huang et al. (2022b), we use the most probable path $\hat{A}$ to sample the latent variable Y_t to avoid performing multiple forward passes through the neural network for all paths in the DAG in order to compute the objective.

The objective at time step t is

$$\mathcal{L}_t = \mathbb{E}_{Q(Y_t|Y_0,\hat{A})} \sum_A \Big[\log P_\theta(A|Y_t, X) - \mathrm{KL}[Q(Y_{t-1}|Y_t, Y_0, A) \| P_\theta(Y_{t-1}|Y_t, A, X)] \Big].$$

We compute the objective at time step t as follows:

$$\mathcal{L}_t = \sum_A \sum_{i=1}^{M} \gamma_t b_t^i \log \mathbf{P}_{a_i, y_0^i} \sum_{i=2}^{M} \mathbf{E}_{a_{i-1}, a_i}$$

where $b_t^i = \begin{cases} 1 & \text{if} \quad y_t^i = [M] \\ 0 & \text{otherwise} \end{cases}$, and γ_t is the hyper-parameters defined by the forward process transition probability. Please refer to Appendix B for detailed derivation. This objective can be optimized using a dynamic programming algorithm similar to the approach in (Huang et al., 2022b). The key differences between Diff-DAT and DAT lie in the sampling of latent variables and the computation of the objective function based on the sampled latent variables at each step.

Inference: Diff-DAT can reuse various decoding strategies from DAT, such as greedy, lookahead, and joint-Viterbi, to perform decoding in a single iteration. However, unlike DAT, we can perform flexible iterative decoding based on the diffusion backward process. Given the model prediction from the previous step, we can continuously sample the latent variable for the next time step and perform denoising. Unlike the fixed T value during training, the diffusion can predict multiple steps at a time, resulting in a more flexible number of decoding steps during inference.

4 Experiments

Experimental Setup. We conduct experiments on multiple public NMT datasets: IWSLT14 En-De/De-En (Cettolo et al., 2014), WMT14 En-De (Bojar et al., 2014), and WMT16 Ro-En/En-Ro (Bojar et al., 2016). To ensure a fair comparison, we used the same settings as previous works (Ghazvininejad et al., 2019; Huang et al., 2022c) and reported the test performance in BLEU score (Papineni et al., 2002a) (Appnedix A).

Main results: Table 1 demonstrates that Diff-DAT significantly outperforms the baselines while maintaining low decoding latency. Compared to DAT, Diff-DAT shows improvements even in the first iteration, implying that diffusion improves DAT's ability to find the optimal path for each reference, effectively addressing multi-modality. We observed a more pronounced improvement between the first and second steps in the WMT16 En-Ro and IWSLT14 De-En datasets, where we used a smaller graph size ($\lambda = 4$). Given that these datasets exhibit lower multi-modality due to the smaller training data and the search space in the graph is limited, the second step of Diff-DAT remarkably boosts translation quality by capturing more dependencies between tokens. We also observed that the improvement between the two steps is more pronounced when using the "Lookahead" decoding strategy. We argue that this occurs because the path found by "Lookahead" is less optimal than "Joint-Viterbi". As a result, the improvement relies more on the better dependencies captured between tokens. Diff-DAT's second decoding step outperforms FA-DAT, demonstrating the capability of iterative decoding in addressing the monotonic assumption in DAT. Diff-DAT outperforms all other single-step decoding methods and achieves comparable or better performance than other multistep decoding methods while maintaining a minimal trade-off in decoding latency.

Although the performance gains in terms of BLEU score are not significantly higher than FA-DAT (Ma et al., 2023), we would like to highlight some of the limitations of FA-DAT: (i) Firstly, FA-DAT optimizes the n-gram count objective, which closely aligns with how the BLEU metric evaluates translation quality. Our comparison using an alternative metric, such as COMET (in table 2), shows that Diff-DAT significantly outperforms FA-DAT. Furthermore, the improvement of FA-DAT over DAT in terms of COMET is less pronounced than when measured with BLEU, indicating that FA-DAT is particularly effective only under BLEU evaluation. It is also worth noting that Diff-DAT can be combined with FA-DAT to further enhance performance. (ii) Secondly, FA-DAT exhibits behavior similar to Knowledge Distillation (KD), as both methods weaken the model's ability to capture multiple translation modalities in the data, leading to reduced translation diversity. FA-DAT causes the vertex distribution to become less diverse, while KD reduces data modalities, resulting in less diverse translations. In fact, the performance of FA-DAT closely aligns with that of DAT combined with KD (demonstrated in Huang et al. (2022b)), and combining FA-DAT with KD further degrades performance (demonstrated in Ma et al. (2023)). Although FA-DAT and KD improve performance

Table 1: Results on WMT14 En-De, WMT16 En-Ro, WMT17 Zh-En and IWSLT14 En-De/De-En datasets. The best performance of the NAT methods is **bolded**. * denotes our implementation.

Model		Iter.	Speedup	IWSLT14 De-En	WMT14 En-De	WMT16 En-Ro	WMT17 Zh-En
Transformer (Vaswani et al., 2017b)		M	1.0×	34.66	27.60	34.16	23.70
CMLM (Ghazvininejad et al., 2019)		10	2.2×	31.80	24.61	32.86	-
CMLM+SMART (Ghazvininejad et al., 2020b)		10	2.2×	30.74	25.10	32.71	-
DiSCo (Kasai et al., 2020)		≈ 4	3.5×	-	25.64	-	-
Imputer (Saharia et al., 2020)		8	2.7×	-	-	25.00	-
CMLMC (Huang et al., 2022c)		10	1.7×	34.28	26.40	34.14	-
DAT *(Huang et al., 2022b)	+ Lookahead	1	14.0×	33.79	26.52	33.46	22.42
	+ Joint-Viterbi	1	13.2×	34.02	26.67	33.65	23.00
FA-DAT * (Ma et al., 2023)	+ Lookahead	1	14.0×	34.65	27.29	33.72	22.73
	+ Joint-Viterbi	1	13.2×	34.66	**27.31**	33.74	22.87
Diff-DAT	+ Lookahead	1	14.0×	34.21	26.34	33.65	22.60
		2	11.8×	34.68	26.83	**34.01**	22.82
	+ Joint-Viterbi	1	13.2×	34.37	26.94	33.72	23.60
		2	9.2×	**34.77**	27.12	34.00	**23.78**

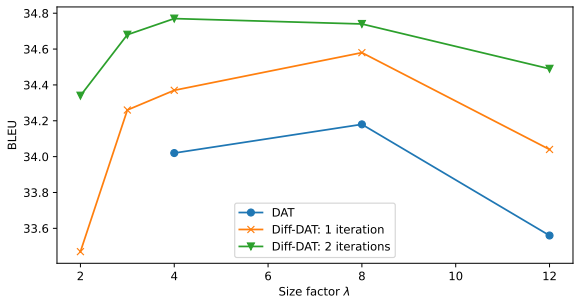


Figure 1: Effects of λ on IWSLT14 De-En. The graph size of DAT is λ times the of source length. We use Join-Viterbi decoding to evaluate BLEU.

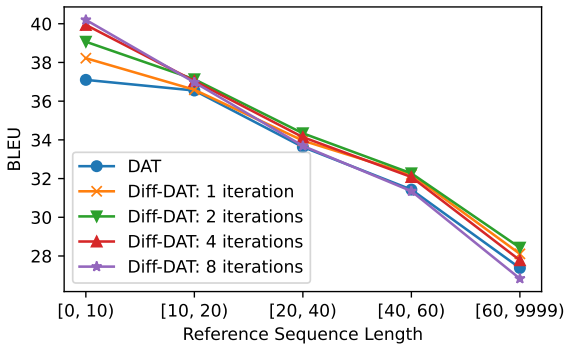


Figure 2: The effect of sequence length.

when measured specifically by BLEU, they overlook the most fundamental challenge in NAT—the multi-modality problem. This limitation causes the model to perform poorly when evaluated with alternative metrics (e.g., COMET) and diversity measures (e.g., pairwise BLEU). We also present a combination of Diff-DAT and FA-DAT, which we refer to as FA-Diff-DAT, in Appendix C.

Ablation study on the graph size: Figure 1 illustrates the results of DAT and Diff-DAT with varying graph sizes. As noted by Huang et al. (2022b), larger graph sizes complicate transition predictions, leading to a performance decline for both DAT and Diff-DAT. Nonetheless, Diff-DAT consistently outperforms DAT across all graph sizes. The graph size also influences the iterative decoding performance of Diff-DAT. Both transition predictions and iterative refinement contribute to capturing token dependencies in the generated sentence. When the graph size is small, iterative decoding significantly aids due to the limitations in transition predictions.

Ablation study on the number of decoding steps: We examine the impact of graph size on iterative decoding. Figure 2 displays BLEU scores categorized by reference length on IWSLT14 DE-EN dataset, results on other datasets are shown in the Appendix. Contrasting trends were observed in different length intervals: in the first interval (length [0, 40)), the BLEU score increases as the number of decoding iterations increases; in the second interval (length [40, ∞)), the BLEU score decreases as the number of decoding iterations increases. Since sequence length directly influences graph size, we conclude that once the graph size reaches a certain threshold, it negatively affects iterative decoding. Larger graph sizes make transition probability prediction more challenging. Additionally, each decoding iteration generates sub-paths in the graph that the model must navigate, further complicating the prediction of transition probabilities. These fac-

Model	IWSLT14 De-En BLEU	IWSLT14 De-En COMET	WMT16 En-Ro BLEU	WMT16 En-Ro COMET
DAT	34.02	0.7617	33.65	0.7608
FA-DAT	34.66	0.7637	33.74	0.7618
Diff-DAT - 1 iter	34.37	0.7642	33.72	0.7619
Diff-DAT - 2 iters	34.77	0.7720	34.00	0.7717
FA-Diff-DAT - 1 iter	34.73	0.7655	34.13	0.7645
FA-Diff-DAT - 2 iters	35.06	0.7727	34.27	0.7751

Table 2: Comparison between Diff-DAT, DAT, and FA-DAT on COMET metric (Rei et al., 2020).

tors make it increasingly difficult for the model to predict transitions with each iteration. Errors from previous iterations affect subsequent ones, further damaging the model's ability to capture dependencies through transition predictions. This intriguing challenge is left for future work.

5 Conclusion

Our study introduces Diff-DAT, a novel approach for enhanced non-autoregressive machine translation. Through a fusion of diffusion models and DAT objectives and the integration of various decoding schemes, Diff-DAT effectively addresses the multi-modality problem, achieving superior translation quality while maintaining fast decoding speed. Extensive experiments across diverse benchmarks demonstrate the effectiveness of Diff-DAT, establishing a new state-of-the-art in non-autoregressive translation. Our work bridges the gap between decoding efficiency and translation quality, advancing the field of sequence-to-sequence learning.

6 Limitations

Although the optimal trade-off between decoding latency and performance occurs with 2-step decoding, increasing the number of decoding iterations does not lead to consistent performance improvements. Instead, it degrades the performance, particularly for long sentences, as demonstrated in our ablation study. This presents an ongoing challenge for iterative decoding in DAT, which remains an area for future research.

References

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pages 12–58.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation (wmt16). In *First conference on machine translation*, pages 131–198. Association for Computational Linguistics.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th iwslt evaluation campaign. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 2–17.

Cunxiao Du, Zhaopeng Tu, and Jing Jiang. 2021. Order-agnostic cross entropy for non-autoregressive machine translation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2849–2859. PMLR.

Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. 2020a. Aligned cross entropy for non-autoregressive machine translation. In *International Conference on Machine Learning*, pages 3515–3523. PMLR.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.

Marjan Ghazvininejad, Omer Levy, and Luke Zettlemoyer. 2020b. Semi-autoregressive training improves mask-predict decoding. *CoRR*, abs/2001.08785.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Shangtong Gui, Chenze Shao, Zhengrui Ma, Yunji Chen, Yang Feng, et al. 2024. Non-autoregressive machine translation with probabilistic context-free grammar. *Advances in Neural Information Processing Systems*, 36.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Chenyang Huang, Hao Zhou, Osmar R. Zaïane, Lili Mou, and Lei Li. 2022a. Non-autoregressive translation with layer-wise prediction and deep supervision. *The Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*.

Fei Huang, Hao Zhou, Yang Liu, Hang Li, and Minlie Huang. 2022b. Directed acyclic transformer for non-autoregressive machine translation. In *International Conference on Machine Learning*, pages 9410–9428. PMLR.

Xiao Shi Huang, Felipe Perez, and Maksims Volkovs. 2022c. Improving non-autoregressive translation models without distillation. In *International Conference on Learning Representations*.

Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. 2020. Non-autoregressive machine translation with disentangled context transformer. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5144–5155. PMLR.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Zhengrui Ma, Chenze Shao, Shangtong Gui, Min Zhang, and Yang Feng. 2023. Fuzzy alignments in directed acyclic graph for non-autoregressive machine translation. In *International Conference on Learning Representations*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 48–53. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002a. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002b. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

L. Qian, H. Zhou, Y. Bao, M. Wang, L. Qiu, W. Zhang, Y. Yu, and L. Li. 2021a. Glancing transformer for non-autoregressive neural machine translation. In *Meeting of the Association for Computational Linguistics*.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021b. Glancing transformer for non-autoregressive neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1993–2003, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. *arXiv preprint arXiv:2009.09025*.

Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. 2020. Non-autoregressive machine translation with latent alignments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1098–1108, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Chenze Shao, Zhengrui Ma, and Yang Feng. 2022. Viterbi decoding of directed acyclic transformer for non-autoregressive machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4390–4397, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. *Advances in neural information processing systems*, 30.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural*

Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008.

A Details Experimental Setup

Table 3: The number of sentence pairs of the training, validation, and test sets in each dataset

	IWSLT14 De-En	WMT14 En-De	WMT16 En-Ro	WMT17 Zh-En
Train	160k	4.5M	610k	20M
Validation	6.75k	3k	2k	2k
Test	6.75k	3k	2k	2k

Table 4: Hyper-parameters

Hyper-parameters	IWSLT14 De-En	WMT14 En-De	WMT16 En-Ro	WMT17 Zh-En
learning rate	0.0005	0.0005	0.0005	0.0005
warmup	30k	10k	15k	10k
dropout	0.3	0.1	0.3	0.1
updates	200k	300k	200k	300k
batch size	8k tokens	64k tokens	32k tokens	64k tokens
Size factor λ	4	8	4	8

Datasets. We conduct experiments on four popular machine translation dataset WMT14 English-German (En-De, 4.5M sentence pairs), WMT17 Chinese-English (Zh-En, 20M sentence pairs) and WMT16 English-Romanian (En-Ro, 610k sentence pairs) and IWSLT14 German-English (De-En 160k sentence pairs). The details size of each dataset is given in table 3. We apply BPE (Sennrich et al., 2015) to learn a joint subword vocabulary for En-De, En-Ro, and De-En and separate vocabularies for Zh-En on the tokenized data.

Baselines. We compare our Diff-DAT against leading NAR baselines, including CMLM (Ghazvininejad et al., 2019) and its variants, CMLM+SMART (Ghazvininejad et al., 2020b), CMLM+AXE (Ghazvininejad et al., 2020a) and CMLM+OaXE (Du et al., 2021), DisCo (Kasai et al., 2020), Imputer (Saharia et al., 2020), GLAT (Qian et al., 2021b), DSLP (Huang et al., 2022a), CMLMC (Huang et al., 2022c), PCFG-NAT (Gui et al., 2024), DAT (Huang et al., 2022b), and FA-DAT (Ma et al., 2023).

Metrics. For fair comparisons with previous work, we use tokenized BLEU (Papineni et al., 2002b) for all benchmarks. The decoding speedup is measured with a batch size of 1.

Implementation details. All baselines and our proposed method are implemented using the open-source toolkit Fairseq (Ott et al., 2019). BLEU scores are evaluated on the validation set, and the final model is obtained by averaging the best 5 checkpoints. To ensure fair comparisons with previous work, we adhere to the training hyper-parameters set by (Huang et al., 2022b,c), as detailed in Table 4. On the IWSLT14 dataset, we employ the Transformer-*small* configuration 512-1024-4, whereas on the WMT datasets, we utilize the Transformer-*base* configuration 512-2048-8 for both the encoder and decoder in our autoregressive baseline. These numerical values correspond to the embedding dimension, FFN layer size, and number of attention heads, respectively. Our model architecture strictly adheres to the settings of DAT, where we set the decoder length to 8 times the source length ($\lambda = 8$) and incorporate graph positional embeddings as decoder inputs unless otherwise specified. Additionally, we set the number of time steps adaptively equal to the length of the source sentence. We assess BLEU scores on the validation set and average the best 5 checkpoints to obtain the final model. In cases where DAT performance is not reported for the WMT16 dataset, we independently train the model using the original code and maintain the same settings in our Diff-DAT approach. Throughout all experiments, we utilize the Adam optimizer (Kingma and Ba, 2014) with default settings and conduct training on 4 Nvidia A100-80G GPUs.

B Derivations of the Variational Lower Bound for Diff-DAT

The following provides the derivation for the loss objective of Diff-DAT:

$$
\begin{aligned}
&\log P_\theta(Y_0|X) = \log \sum_{Y_{1:T}} \sum_{A} P_\theta(Y_{0:T}, A|X) \\
&= \log \sum_{A} \mathbb{E}_{Q(Y_{1:T}|Y_0,A)} \frac{P_\theta(Y_{0:T}, A_{1:T}|X)}{Q(Y_{1:T}|Y_0, A)} \\
&\geq \sum_{A} \mathbb{E}_{Q(Y_{1:T}|Y_0,A)} \log \frac{P_\theta(Y_{0:T}, A_{1:T}|X)}{Q(Y_{1:T}|Y_0, A)} \\
&= \sum_{A} \mathbb{E}_{Q(Y_{1:T}|Y_0,A)} \log \frac{P_\theta(Y_T|X) \prod_{t=1}^{T} P_\theta(A|Y_t, X) P_\theta(Y_{t-1}|Y_t, A, X)}{Q(Y_T|Y_0) \prod_{t=2}^{T} Q(Y_{t-1}|Y_t, Y_0, A)} \\
&= \sum_{t=2}^{T} \mathbb{E}_{Q(Y_t|Y_0,A)} \sum_{A} \Big[\log P_\theta(A|Y_t, X) - \mathrm{KL}[Q(Y_{t-1}|Y_t, Y_0, A) \| P_\theta(Y_{t-1}|Y_t, A, X)] \Big] \\
&+ \mathbb{E}_{Q(Y_1|Y_0)} \sum_{A} \Big[\log P_\theta(A|Y_1, X) + \log P_\theta(Y_0|Y_1, A, X) \Big] + \text{const}
\end{aligned}
$$

Forward process: At time step t, for each path A_t in the DAG, we sample $Y_t \sim Q(Y_t|Y_{t-1}, A_t)$ by applying the following forward transition probabilities independently for each token y^i_{t-1} in the sequence Y_{t-1}:

$$
\begin{cases}
Q(y^i_t = [M]|y^i_{t-1} = [M], a_i \in A) & = & 1 \\
Q(y^i_t = y^i_0|y^i_{t-1} = y^i_0, a_i \in A) & = & \beta_t \\
Q(y^i_t = [M]|y^i_{t-1} = y^i_0, a_i \in A) & = & 1 - \beta_t \\
Q(y^i_t = y^i_0|y^i_{t-1} = [M], a_i \in A) & = & 0 \\
Q(y^i_t = [M]|y^i_{t-1}, a_i \notin A) & = & 1 \\
Q(y^i_t \neq [M]|y^i_{t-1}, a_i \notin A) & = & 0
\end{cases}
\tag{7}
$$

For each token, if it is part of the current path A_t, it has a probability of transitioning to the absorbing state $[M]$, or it may remain unchanged. Tokens that are not part of the path, or were already masked in previous steps, will always remain in the absorbing state. We can also perform a t-step marginal, $Q(Y_t|A_t, Y_0)$, to directly sample Y_t from Y_0:

$$
\begin{cases}
Q(y^i_t = y^i_0|y^i_0, a_i \in A) & = & \alpha_t \\
Q(y^i_t = [M]|y^i_0, a_i \in A) & = & 1 - \alpha_t \\
Q(y^i_t = [M]|y^i_0, a_i \notin A) & = & 1 \\
Q(y^i_t = y^i_0|y^i_0, a_i \notin A) & = & 0
\end{cases}
\tag{8}
$$

where $\alpha_t := \prod_{i=1}^{t} \beta_t$ is specified to decrease from 1 to 0

Backward process: We then compute the posterior at time $t-1$ as $Q(Y_{t-1}|Y_t, Y_0, A) = \frac{Q(Y_t|Y_{t-1},A)Q(Y_{t-1}|Y_0,A)}{Q(Y_t|Y_0,A)}$. The backward transition probabilities for each token are then calculated as follows:

$$
\begin{cases}
Q(y^i_{t-1} = y^i_0|y^i_t = [M], y^i_0, a_i \in A) & = & \gamma_t \\
Q(y^i_{t-1} = [M]|y^i_t = [M], y^i_0, a_i \in A) & = & 1 - \gamma_t \\
Q(y^i_{t-1} = [M]|y^i_t = y^i_0, y^i_0, a_i \in A) & = & 0 \\
Q(y^i_{t-1} = [M]|a_i \notin A) & = & 1
\end{cases}
\tag{9}
$$

where $\gamma_t = \frac{\alpha_{t-1} - \alpha_t}{1 - \alpha_t}$. Instead of training a neural network $f_\theta(Y_t, X)$ to directly predict the logits of the distribution $P_\theta(Y_{t-1}|Y_t, A, X)$, we follow the approach in (Ho et al., 2020) and focus on learning a neural network $f_\theta(Y_t, X)$ to predict the logits of the distribution $\tilde{P}_\theta(\tilde{Y}_0|Y_t, A, X)$. We then combine this with $Q(Y_{t-1}|Y_t, Y_0, A)$ and sum over the one-hot representations of Y_0 to obtain the following parameterization: $P_\theta(Y_{t-1}|Y_t, A, X) = \sum_{\tilde{Y}_0} \tilde{P}_\theta(\tilde{Y}_0|Y_t, A, X) Q(Y_{t-1}|Y_t, \tilde{Y}_0, A)$ The parameterized backward transition probabilities for each token are then derived as follows:

$$\begin{cases} P_\theta(y^i_{t-1} = y^i_0 | y^i_t = [M], a_i \in A, X) = \gamma_t \mathbf{P}_{a_i, y^i_0} \\ P_\theta(y^i_{t-1} = [M] | y^i_t = [M], a_i \in A, X) = 1 - \gamma_t \\ P_\theta(y^i_{t-1} = [M] | y^i_t = y^i_0, a_i \in A, X) = 0 \\ P_\theta(y^i_{t-1} = [M] | a_i \notin A, X) = 1 \end{cases} \tag{10}$$

where $\mathbf{P} = \tilde{P}_\theta(\tilde{y}^i_0 | Y_t, X) = f_\theta(Y_t, X) \in \mathbb{R}^{L \times |V|}$ is the matrix containing the token distributions on the L vertices, and $\tilde{P}_\theta(\tilde{y}^i_0 = y^i_0 | a_i, Y_t, X) = \mathbf{P}_{a_i, y^i_0}$. Since the forward and backward processes are factorized as conditionally independent over the image or sequence elements, the KL divergence between Q and P_θ can be computed by simply summing over all possible values of each random variable, which is given by the following:

$$\begin{aligned} &\mathrm{KL}(Q(y^i_{t-1} | y^i_t, y^i_0, A_t) \| P_\theta(y^i_{t-1} | y^i_t, A, X)) = \\ &\begin{cases} -\gamma_t \log \mathbf{P}_{a_i, y^i_0} & \text{if} \quad y^i_t = [M] \text{ and } a_i \in A \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{11}$$

Put it into $\mathcal{L}_t$ in (6), we have the objective:

$$\mathcal{L}_t = \sum_A \mathbb{E}_{Q(Y_t|Y_0, A)} \sum_{i=1}^{M} \gamma_t b^i_t \log \mathbf{P}_{a_i, y^i_0} \sum_{i=2}^{M} \mathbf{E}_{a_{i-1}, a_i}$$

where $b^i_t = \begin{cases} 1 & \text{if} \quad y^i_t = [M] \\ 0 & \text{otherwise} \end{cases}$. Optimizing requires multiple forward passes of the neural network each time we sample $Y_t \sim Q(Y_t | Y_0, A)$ given a path A. To simplify this, we condition only on the path with the highest probability $\hat{A} = \arg\max_A P_\theta(Y_0, A | X)$, and sample $Y_t \sim Q(Y_t | Y_0, \hat{A})$. Consequently, $\mathcal{L}_t$ becomes:

$$\mathcal{L}_t = \mathbb{E}_{Q(Y_t|Y_0, \hat{A})} \sum_A \sum_{i=1}^{M} \gamma_t b^i_t \log \mathbf{P}_{a_i, y^i_0} \sum_{i=2}^{M} \mathbf{E}_{a_{i-1}, a_i}$$

We can efficiently compute $\mathcal{L}_t$ using dynamic programming, similar to the approach in DAT. During training, we sample a time step t and update the model parameters to maximize the objective in (**??**).

C Fuzzy Alignment in Diff-DAT

FA-DAT (Ma et al., 2023) addresses the strict monotonic alignment in DAT by fine-tuning it using a fuzzy alignment objective as follows:

$$p_n(\tilde{Y}_0, Y_0) = \frac{\sum_{g \in G_n(Y_0)} \min(C_g(\tilde{Y}_0), C_g(Y_0)}{\sum_{g \in G_n(\tilde{Y}_0)} C_g(\tilde{Y}_0)} \tag{12}$$

$$p_n(\theta, Y_0) = \mathbb{E}_{P_\theta(A|X)} p_n(P_\theta(\tilde{Y}_0 | A, X), Y_0) \tag{13}$$

Due to the high complexity, they optimize a more tractable approximation of (13) as follows:

$$\tilde{p}_n(\theta, Y_0) = \frac{\sum_{g \in G_n(Y_0)} \min(\mathbb{E}_{P_\theta(A|X)} C_g(P_\theta(\tilde{Y}_0 | A, X)), C_g(Y_0)}{\mathbb{E}_{P_\theta(A|X)} \sum_{g \in G_n(\tilde{Y}_0)} C_g(P_\theta(\tilde{Y}_0 | A, X))} \tag{14}$$

(14) can be efficiently computed using Dynamic Programming as outlined in (Ma et al., 2023). We extend this objective for fine-tuning Diff-DAT as follows:

$$\tilde{p}_n(\theta, Y_0, t) = \frac{\sum_{g \in G_n(Y_0)} \min(\mathbb{E}_{Q(Y_t|Y_0, \hat{A}, X)} \mathbb{E}_{P_\theta(A|X)} C_g(P_\theta(\tilde{Y}_0 | Y_t, A, X)), C_g(Y_0)}{\mathbb{E}_{Q(Y_t|Y_0, \hat{A}, X)} \mathbb{E}_{P_\theta(A|X)} \sum_{g \in G_n(\tilde{Y}_0)} C_g(P_\theta(\tilde{Y}_0 | Y_t, A, X))} \tag{15}$$

Similar to Diff-DAT, at each gradient descent step, we sample $t \sim \text{Uniform}(T)$ and $Y_t \sim Q(Y_t|Y_0, A^0, X)$ to compute the objective (15).

We conducted experiments to compare Diff-DAT with a Fuzzy Alignment objective (FA-Diff-DAT) against other baselines, as shown in Table 5. When combined with the Fuzzy Alignment objective, FA-Diff-DAT further improves performance, surpassing FA-DAT in both 1 and 2 iterations of decoding.

Table 5: Comparison between Diff-DAT, FA-Diff-DAT, DAT, and FA-DAT. The highest-performing NAT methods are highlighted in **bold**. * indicates our implementation.

Model		**Iter.**	**Speedup**	**IWSLT14 De-En**	**WMT16 En-Ro**
Transformer (Vaswani et al., 2017b)		M	1.0×	34.66	34.16
DAT *(Huang et al., 2022b)	+ Lookahead	1	14.0×	33.79	33.46
	+ Joint-Viterbi	1	13.2×	34.02	33.65
	+ Greedy	1	14.2×	33.68	33.28
FA-DAT * (Ma et al., 2023)	+ Lookahead	1	14.0×	34.65	33.72
	+ Joint-Viterbi	1	13.2×	34.66	33.74
	+ Greedy	1	14.2×	34.64	33.69
Diff-DAT	+ Lookahead	1	14.0×	34.21	33.65
		2	11.8×	34.68	34.01
	+ Joint-Viterbi	1	13.2×	34.37	33.72
		2	9.2×	34.77	34.00
	+ Greedy	1	14.2×	33.96	33.47
		2	12.2×	34.51	33.87
FA-Diff-DAT	+ Lookahead	1	14.0×	34.71	33.74
		2	11.8×	35.04	33.97
	+ Joint-Viterbi	1	13.2×	34.73	34.13
		2	9.2×	**35.06**	**34.27**
	+ Greedy	1	14.2×	34.69	33.72
		2	12.2×	35.03	33.94

D Ablation Study on the Impact of Size Factor λ

Figure 3 illustrates the results of DAT and Diff-DAT with varying graph sizes. Previous work (DAT) indicates that larger graph sizes complicate transition predictions, leading to a performance decline for both DAT and Diff-DAT. Nonetheless, Diff-DAT consistently outperforms DAT across all graph sizes. The graph size also influences the iterative decoding performance of Diff-DAT. Both transition predictions and iterative refinement contribute to capturing token dependencies in the generated sentence. When the graph size is small, iterative decoding significantly aids due to the limitations in transition predictions.

E Analysis of the impact of varying the number of decoding steps

We conducted another analysis to examine the impact of graph size on iterative decoding. Figures 4, 5, 6, and 7 display the BLEU scores for each dataset, categorized by reference length. A consistent trend was observed that in the results of each dataset there are two intervals: in the first interval, the BLEU score increases as the number of decoding iterations increases; in the second interval, the BLEU score decreases as the number of decoding iterations increases. Since sequence length directly influences graph size, it can be concluded that once the graph size reaches a certain threshold, it negatively affects iterative decoding. Larger graph sizes make it more challenging to predict transition probabilities. Additionally, each decoding iteration generates sub-paths in the graph that the model must navigate, further complicating the prediction of transition probabilities. These factors combined make it increasingly difficult for the model to predict transitions with each iteration. Moreover, errors from previous iterations affect subsequent ones, leading

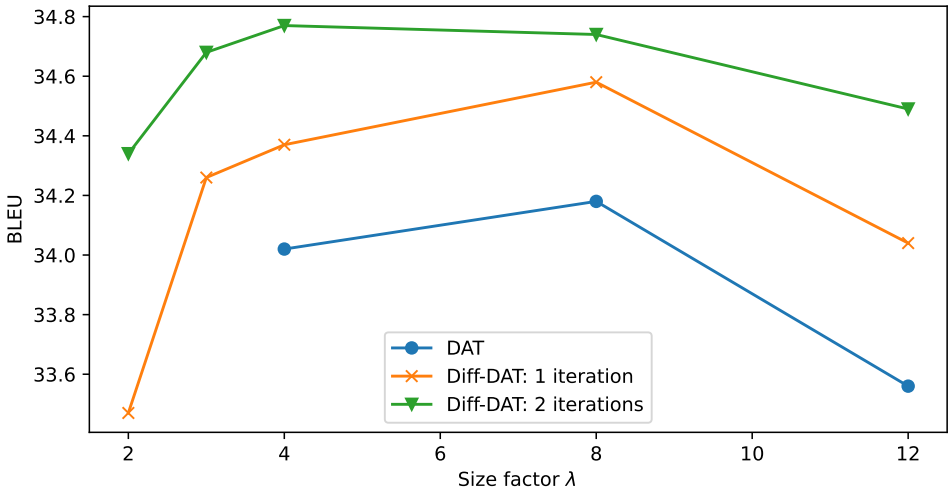


Figure 3: Effects of λ on IWSLT14 De-En. The graph size of DAT is λ times the of source length. We use Join-Viterbi decoding to evaluate BLEU.

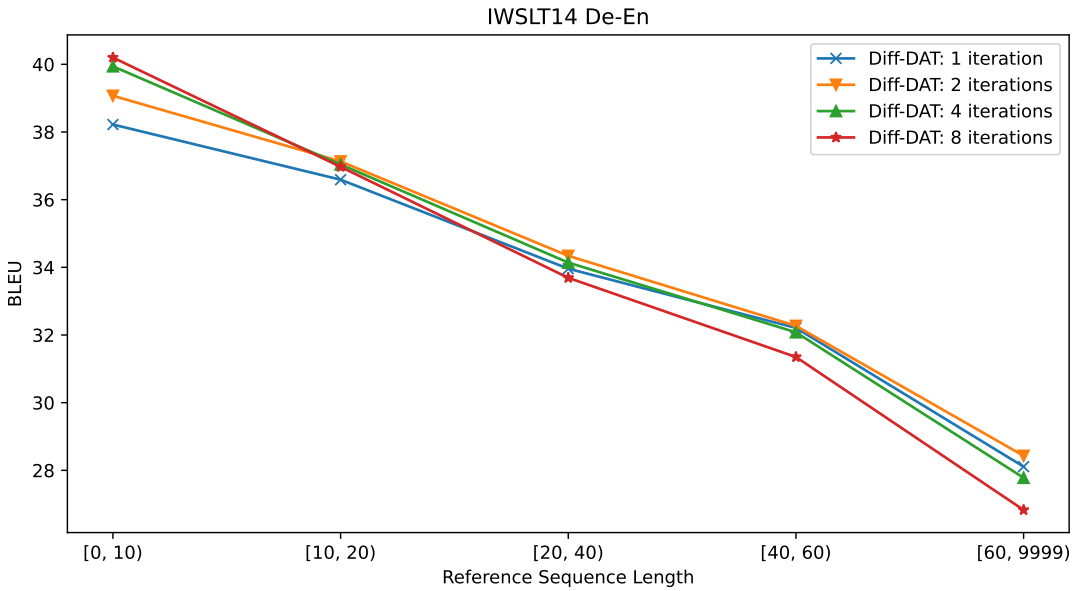


Figure 4: The BLEU score on IWSLT14 De-En bucketed by the reference length.

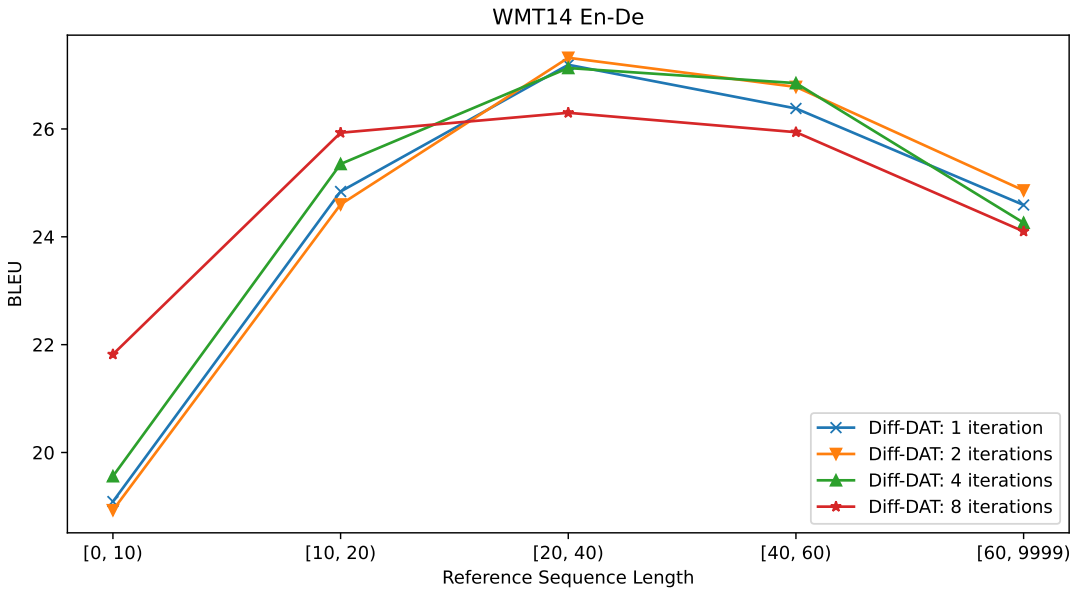


Figure 5: The BLEU score on WMT14 En-De bucketed by the reference length.

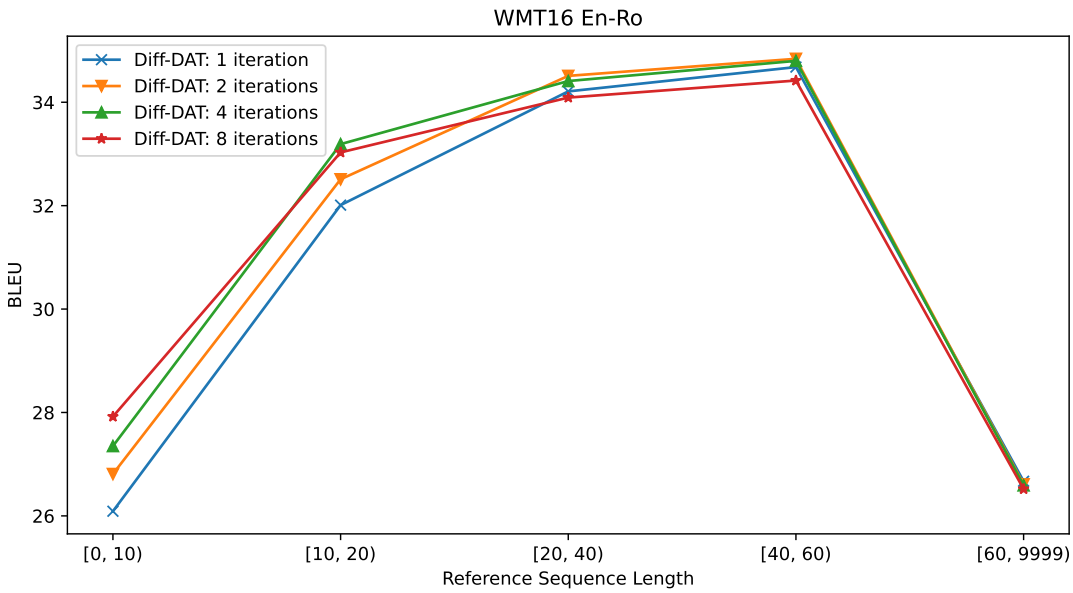


Figure 6: The BLEU score on WMT16 En-Ro bucketed by the reference length.

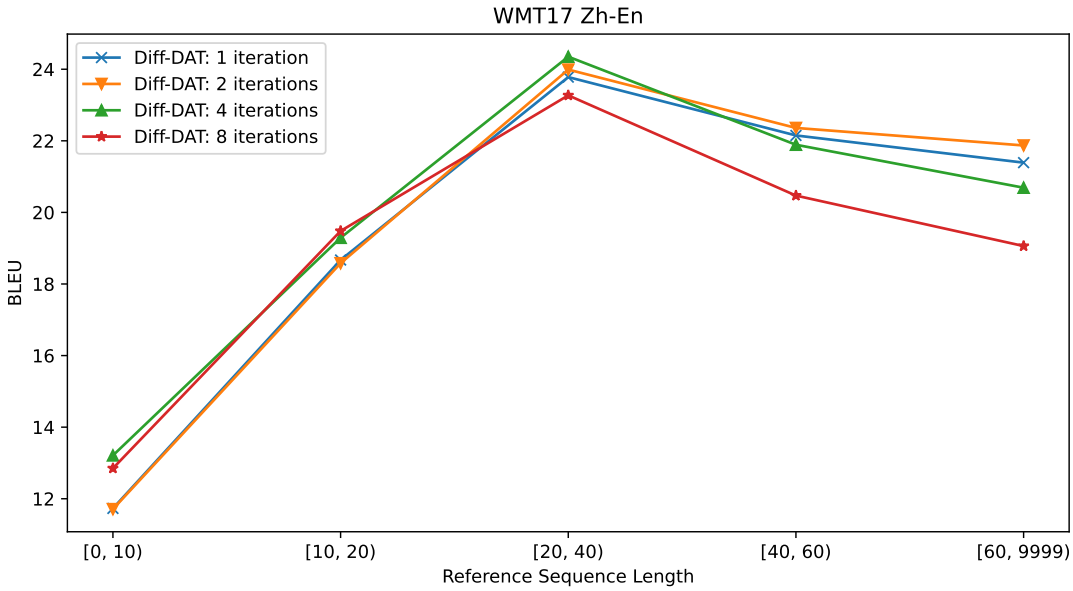


Figure 7: The BLEU score on WMT17 Zh-En bucketed by the reference length.

to a negative impact on the model's ability to capture dependencies within the sentence through transition predictions. This intriguing challenge inspires us to address it in future work.

F Inference

In inference, we adopt the translation of DAG with *Greedy*, *Lookahead* (Huang et al., 2022b), and *Joint-Viterbi* (Shao et al., 2022) decoding.

Greedy The most straightforward approach involves selecting the most probable transitions and tokens. In essence, we conduct simultaneous argmax operations to determine the most probable transition and token for every vertex. Subsequently, we construct the translation by gathering the predicted tokens along the selected path. This greedy decoding method is remarkably efficient, requiring only two parallel operations, as illustrated in Algorithm 1.

Algorithm 1 Greedy / Lookahead Decoding in Pytorch-like Parallel Pseudocode

Input: Graph Size L, Transition Matrix $\mathbf{E} \in \mathbb{R}^{L \times L}$,
if Using Lookahead **then**
  $\mathbf{E} := \mathbf{E} \otimes [\mathbf{P}.\text{MAX}(\text{dim=1}).\text{UNSQUEEZE}(\text{dim=0})]$
  # $\mathbf{E}$ now jointly considers $\mathbf{P}$ and $\mathbf{E}$
  # $\otimes$ is element-wise multiplication
end if
tokens $:= \mathbf{P}.\text{ARGMAX}(\text{dim=1})$ # shape: (L)
edges $:= \mathbf{E}.\text{ARGMAX}(\text{dim=1})$ # shape: (L)
$i := 1$, *output* := [*tokens*[1]]
repeat
  $i :=$ *edges*$[i]$ # jumping along the transition
  output.APPEND(*tokens*$[i]$)
until $i = L$

Lookahead Lookahead decoding enhances the efficacy of the greedy approach by jointly considering both transitions and tokens. Specifically, we rearrange $P_\theta(Y, A|X)$ into the following sequential decision problem:

$$P_\theta(y_1|a_1, X) \prod_{i=2}^{M} P_\theta(a_i|a_{i-1}, X) P_\theta(y_i|a_i, X), \tag{16}$$

This formulation transforms the task into choosing a_i and y_i sequentially. We concurrently derive:

$$y_i^*, a_i^* = \arg\max_{y_i, a_i} P_\theta(y_i|a_i, X) P_\theta(a_i|a_{i-1}, X), \tag{17}$$

This can still be executed in parallel with minimal overhead, as outlined in Algorithm 1.

Joint-Viterbi We additionally utilize Joint-Viterbi decoding (Shao et al., 2022) to determine the global joint optimum of translation and path within a predefined length constraint. Subsequently, we reevaluate these candidates through length normalization to refine their ranking.

It is notable that Joint-Viterbi decoding can be seen as enhancements to Greedy decoding and Lookahead decoding, respectively. While both Greedy and Lookahead decoding methods focus on the immediate probability and select the next token using $\arg\max_{a_i} P_\theta(a_i|X, a_{i-1})$ and $\arg\max_{y_i, a_i} P_\theta(y_i|a_i, X) P_\theta(a_i|a_{i-1}, X)$, respectively, Joint-Viterbi algorithm consider the entire decoding path. They ensure the discovery of the globally optimal solution $\arg\max_A P_\theta(A|X)$ and $\arg\max_{A,Y} P_\theta(A, Y|X)$, respectively.

Efficient Knowledge Editing via Minimal Precomputation

Akshat Gupta[1], **Maochuan Lu**[1] **Thomas Hartvigsen**[2], **Gopala Anumanchipalli**[1]
[1]UC Berkeley, [2]University of Virginia
akshat.gupta@berkeley.edu

Abstract

Knowledge editing methods like MEMIT are able to make data and compute efficient updates of factual knowledge by using a single sentence to update facts and their consequences. However, what is often overlooked is a "precomputation step", which requires a one-time but significant computational cost. The authors of MEMIT (Meng et al., 2022b) originally precompute approximately 44 million hidden vectors per edited layer, which requires a forward pass over 44 million tokens. For GPT-J (6B) (Wang and Komatsuzaki, 2021), this precomputation step takes 36 hours on a single GPU, while it takes approximately 40 hours for Llama2-7B (Touvron et al., 2023). Additionally, this precomputation time grows with model size. In this paper, we show that this excessive computational cost is unnecessary. Knowledge editing using MEMIT and related methods, such as ROME and EMMET (Meng et al., 2022a; Gupta et al., 2024c), can be performed by pre-computing a very small portion of the 44 million hidden vectors. We first present the theoretical minimum number of hidden vector precomputation required for solutions of these editing methods to exist. We then empirically show that knowledge editing using these methods can be done by pre-computing significantly fewer hidden vectors. Specifically, we show that the precomputation step can be done with less than 0.3% of the originally stipulated number of hidden vectors. This saves a significant amount of precomputation time and allows users to begin editing new models within a few minutes.

1 Introduction

Knowledge editing (Yao et al., 2023), or the ability to edit knowledge stored within the parameters of large language models (LLMs), is a topic of growing interest. A specific type of parameter-modifying knowledge editing methods called "locate-then-edit" methods (Yao et al., 2023) allow us to edit any transformer-based LLMs without the need for additional training. The most popular of these methods are MEMIT (Meng et al., 2022b), the first successful method that allows for batched editing, its predecessor ROME (Meng et al., 2022a; Gupta et al., 2024a), which allows only one edit at a time, and EMMET (Gupta et al., 2024c), that generalizes ROME to batched editing.

While these "locate-then-edit" methods do not require additional training, we cannot just start editing a newly launched LLM instantly (Yoon et al., 2024). Each of MEMIT, ROME, and EMMET has a precomputation step where a large number of Wikipedia articles are passed through the model being edited and the intermediate hidden representations of the edited layers are cached. The editing loss function aims to preserve the outputs of these cached hidden representations during the editing process (Gupta et al., 2024c). Although this needs to be done only once to edit a model, it still requires a significant computational overhead. For example, the original authors of MEMIT pre-computed about 44 million hidden vectors per edited layer. This computation takes about 36 hours for GPT-J (6B) (Wang and Komatsuzaki, 2021) and 40 hours for Llama-2 (7B) (Touvron et al., 2023) on a single GPU[1]. Additionally, these numbers increase with the size of the model and the number of layers being edited. This means that while locate-then-edit methods do not require additional training and can be very quick during inference, they do require a significant initial computational cost which grows with the model size.

In this paper, we show that this large amount of precomputation is unnecessary. We first analyze the closed-form solution for the different editing algorithms and find the theoretical minimum amount of tokens required for the precomputation

[1]Numbers calculated on a single NVIDIA A6000 GPU with 48 GB GPU memory.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 829–840
July 27 - August 1, 2025

step. We then empirically show that optimal knowledge editing performance can be achieved by performing precomputation on approximately twice this minimum number. This allows us to achieve comparable knowledge editing performance using less than 0.1% of the originally stipulated 44 million tokens for GPT2-XL and GPT-J. We call the efficient versions of these methods as the FastMEMIT family of editing methods, which significantly reduce the upfront computation costs, making it possible to begin editing models within minutes. We also release our code, which can be found here - https://github.com/scalable-model-editing/efficient-model-editing.

2 Background

In "locate-then-edit" knowledge editing methods, facts for model editing are usually represented in a key-value format, where the key vector helps locate a fact, and the value vector provides the target output after editing (Meng et al., 2022b). For example, for the edited fact "The capital of Malaysia is Singapore," k_e corresponds to the query "The capital of Malaysia is," and v_e corresponds to the new target "Singapore." Additionally, k_0 represents key vectors whose outputs need to remain constant during editing, ensuring the editing process doesn't impact the general ability (Gupta et al., 2024b) or unrelated knowledge of edited models.

During editing, we first identify the layer that is maximally responsible for retrieving a fact, and then update the corresponding weight matrix to reflect the updated fact. In this process, we want to make sure two things: one is to preserve previously stored knowledge, and the other one is to memorize what we edit. For MEMIT (Meng et al., 2022b), causal mediation analysis showed that the MLP modules within certain layers are responsible for storing factual knowledge. The knowledge editing objective of MEMIT is formulated as follows (Gupta et al., 2024c):

$$\underset{\hat{W}}{\text{argmin}} \; \underbrace{\lambda \left\| \hat{W}K_0 - W_0K_0 \right\|_F^2}_{\text{preservation}} + \underbrace{\left\| \hat{W}K_E - V_E \right\|_F^2}_{\text{memorization}} \tag{1}$$

The above loss can be interpreted as a summation of two terms. In the first term, we preserve the outputs for a collection of input key-vectors (K_0) to preserve the existing knowledge of the model, while in the second term we force the outputs of certain key-vectors (K_E) to a target (V_E). The argument $\hat{W}$ is the second MLP matrix in the FFN module of a transformer. Since the above objective is linear in the argument, we can derive a closed form solution, as shown below:

$$\hat{W} = W_0 + \Delta \quad \text{where}$$
$$\Delta = \left(V_E - W_0K_E\right)K_E^T\left(\lambda C_0 + K_EK_E^T\right)^{-1} \tag{2}$$

where W_0 is the unedited weight matrix, and $\hat{W}$ refers to the updated weights. k_0 denotes the key-vector for preserving knowledge from the original model. $K_0 = [k_1^0 \,|k_2^0\, | \ldots |\, k_P^0]$ is a matrix containing all these preserved key-vectors. k_e denotes the key-vectors representing modified facts, and $K_E = [k_1^e \,|k_2^e\, | \ldots |\, k_B^e]$ is a matrix containing edited key-vectors. The output at the edited layer corresponding to k_e is denoted by v_e and $V_E = [v_1^e \,|v_2^e\, | \ldots |\, v_B^e]$ is the matrix containing all target vectors.

2.1 Overview of knoweldge editing metrics

In this paper, we evaluate knowledge editing methods using the following standard knowledge editing metrics (Meng et al., 2022b):

- **Efficacy Score (ES)** evaluates the success of an edit. It is calculated as the percentage of edits for which $P(\text{new fact}) > P(\text{old fact})$.
- **Paraphrase Score (PS)** evaluates the model's generalization ability for an edit, calculated as the $P(\text{new fact}) > P(\text{old fact})$ when a paraphrase of the editing prompt is used as query.
- **Neighborhood Score (NS)** evaluates the locality or specificity of an edit. It is calculated as the percentage of the facts in the neighborhood of the edited fact that remain unchanged after an edit.
- **Overall Score (S)** is the harmonic mean of ES, PS, and NS.

3 Dataset and Models

We perform singular and batch editing experiments on the CounterFact dataset (Meng et al., 2022a). CounterFact is a standard dataset used in knowledge editing. We peform knowledge editing on three representative models - GPT2-XL (Radford et al., 2019), GPT-J (6B) (Wang and Komatsuzaki, 2021) and Llama2-7B (Touvron et al., 2023).

4 Theoretical Minimum Tokens for Precomputation

One major benefit of the closed-form solution in MEMIT is the presence of the covariance matrix, $C_0 = K_0 K_0^T$, which can be written as a sum of outer products of key-vectors as shown below:

$$C_0 = K_0 K_0^T = \sum_{i=1}^{P} k_0^i k_0^{i^T} \tag{3}$$

Here, P denotes the number of preserved vectors in equation 1. This matrix C_0 remains fixed during editing since it is made up of key-vectors that serve as the input of the edited matrix, which is why C_0 is precomputed before editing begins. C_0 is one part of the matrix that gets inverted in the closed-form solution of MEMIT (equation 2). If we represent the matrix being inverted in the closed form solution as C_{eff}, then:

$$\begin{aligned} C_{\text{eff}} &= \lambda K_0 K_0^T + K_E K_E^T \\ &= \lambda \sum_{i=1}^{P} k_0^i k_0^{i^T} + \sum_{i=1}^{B} k_e^i k_e^{i^T} \end{aligned} \tag{4}$$

A pre-requisite of the closed-form solution to exist is the invertibility of the C_{eff} matrix. As shown above, C_{eff} matrix is a sum of outer products of $P + B$ vectors, where B represents the batch size for editing. For a model with hidden dimension d, the dimensionality of a key-vector is usually $4d$. This means that the C_{eff} matrix is a square matrix of dimensionality $4d$. For a $4d$-dimensional square matrix which is a summation of rank-1 matrices, it is invertible as long as there are at least $4d$-independent vectors in the summation. For example, for GPT2-XL with hidden dimension of 1600, the dimensionality of key vectors are 6400. Thus, as long as representations of at least 6400 independent key-vectors are preserved or memorized while editing, C_{eff} will be an invertible matrix. *This is a fundamental assumption in MEMIT.*

We want to find the minimum number of keys that need to be preserved in order for C_{eff} to be invertible. Since the editing batch size (B) is varied from one to larger batch sizes, we take $B = 1$ for this argument. If we let the dimensionatity of the key-vectors be d_k, then for an editing batch size of 1, **at least $d_k - 1$ key-vectors need to be preserved, granted they are independent of each other**. This number serves as the theoretical minimum number of tokens over which we need to perform precomputations.

In practice, MEMIT preserves the representations of a much larger number of vectors - 44 million tokens to be specific. For each layer being edited, this step takes about 1.5 hours for GPT-XL, 6 hours for GPT-J, 8 hours for Llama-2-7B. Since multiple layers are edited within a model in MEMIT, this number usually requires tens of hours of precomputation, and scales linearly with the size of the model being edited[2].

5 FastMEMIT Family of Methods

In the above section, we show that the minimum number of tokens required for the computation of C_0 matrix is $d_k - 1$, where d_k is the dimensionality of the key-vectors in an MLP. For GPT2-XL, $d_k = 6400$, whereas for GPT-J, $d_k = 16384$. While the theoretical minimum number of precomputations required is approximately equal to d_k, we ask the question - **"what is the optimal number of tokens required for precomputation without compromising on editing performance?"**.

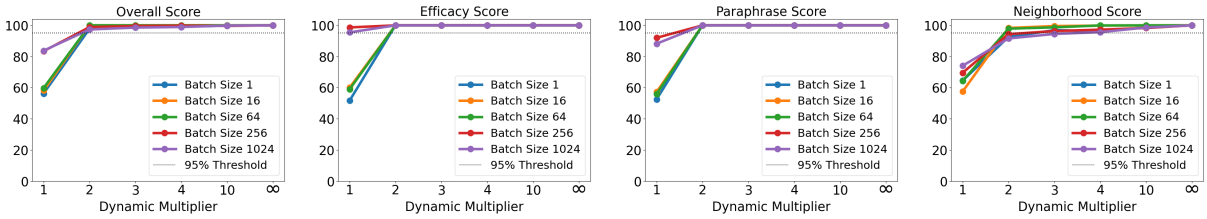
We begin by using the theoretical minimum number for precomputation and quickly find that this leads to loss of editing performance. We also find that for some cases, especially for Llama2-7B models, using this theoretical minimum leads to uninvertible matrices, since the selected vectors may not be independent. We increase the number of precomputation tokens in increments of the theoretical minimum. For this, we introduce **dynamic multiplier**, a hyperparameter that controls the number of preserved key vectors in C_0. For example, with a dynamic multiplier of $d_m = 3$, the number of precomputed key vectors is reduced to $3 \times d_k$, where d_k is approximately equal to the theoretical minimum. This is a significantly lower computational cost while ensuring the matrix remains invertible. For example, with $d_m = 3$ for GPT2-XL, the precomputation is done over 12,288 tokens, which is approximately 0.02% of the original 44 million tokens.

With this dynamic multiplier, we can rewrite Equation 4 as follows:

$$\begin{aligned} C_{\text{eff}} &= \lambda K_0 K_0^T + K_E K_E^T \\ &= \lambda \sum_{i=1}^{P'} k_0^i k_0^{i^T} + \sum_{i=1}^{B} k_e^i k_e^{i^T}, \end{aligned} \tag{5}$$

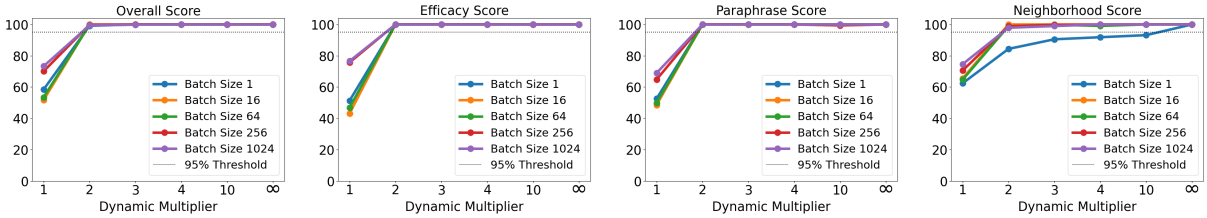
where $P' = d_m \cdot d_k$. The same idea of using the dynamic multiplier to reduce the number of

[2]Numbers calculated for 1 RTX A6000 48GB GPU



(a) Overall Score (b) Efficacy score (c) Paraphrase score (d) Neighborhood score

Figure 1: Performance of FastEMMET in GPT-J across different batch sizes



(a) Overall Score (b) Efficacy score (c) Paraphrase score (d) Neighborhood score

Figure 2: Performance of FastMEMIT in GPT-J across different batch sizes

preserved keys can be applied to ROME (Meng et al., 2022a) and its batch generalization EMMET (Gupta et al., 2024c). We refer to the reduced precomputation version of these methods as FastMEMIT, FastROME, and FastEMMET in this paper.

To evaluate these methods, we perform batched knowledge editing for varying batch sizes, growing from 1 to 1024. For each batch size, we take samples of multiple batches (Table 1 in appendix). For example, for batch size 16, the results are calculated by averaging editing results of 10 batches. Since EMMET is a batch-editing generalization of ROME, we present the results for EMMET in this paper. The editing results for ROME correspond to EMMET with batch size 1.

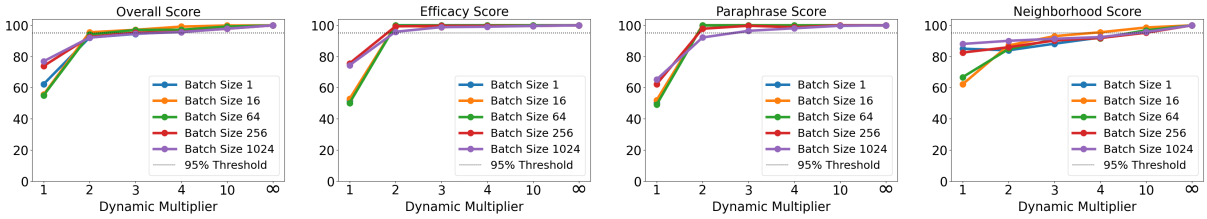
5.1 Results

The knowledge editing results for GPT-J (6B) with reduced precomputation are shown in Figures 1 and 2 for EMMET and MEMIT respectively. The results for the original EMMET and MEMIT algorithm with complete precomputation of 44 million tokens are represented on the x-axis by an "∞" symbol. We present the results for different batch sizes from 1 to 1024 and the different evaluation metrics discussed in section 2.1. The figures also contain a 95% threshold line, which represents 95% performance with respect to the full precomputation value. The exact numerical values for these figures are shared in Appendix A.

We can see that both FastEMMET and FastMEMIT achieve performance that is similar to or even better than the original algorithms with full precomputation, as shown by the overall score metric plots for both EMMET (Figure 1a) and MEMIT (Figure 2a) for GPT-J. The results for GPT2-XL follow a very similar trend and are presented in the appendix (Figures 5 and 6).

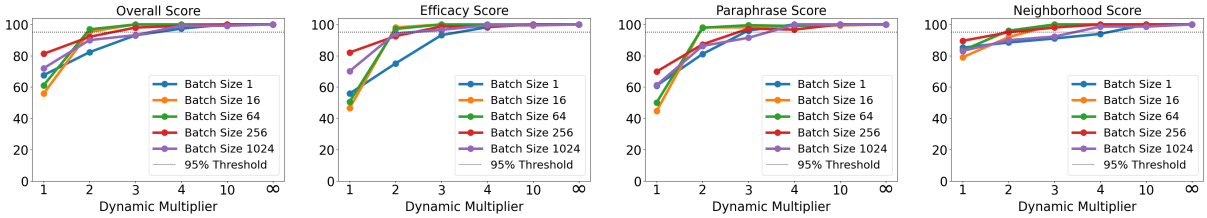
This is true despite using a significantly lower amount of precomputation. Starting at a dynamic multiplier of 2, the editing results are nearly identical to those of the original algorithms where computation is done over 44 million tokens. A dynamic multiplier of 2 means doing precomputation over 32k tokens for GPT-J, which is less than 0.08% of the amount of precomputation required by the original algorithms. For GPT2-XL, a dynamic multiplier of 2 requires precomputation over 12.8k tokens, or 0.02% of the original amount. This enables precomputation to finish within a few seconds, avoiding the large precomputation stage that precedes knowledge editing.

The results for Llama2-7B are shown in Figures 3 and 4. We see that the performance is within the 95% threshold for EMMET even when $d_m = 2$, but MEMIT requires extra precomputation tokens to achieve comparable performance for smaller batch sizes. For MEMIT, we also observe that for smaller batch sizes from 1 to 10, the C_{eff} is not invertible at low values of dynamics multiplier, suggesting that the cached hidden representations are highly correlated. We fix this with a minor regularization term which is added into the closed-form solution



(a) Overall Score (b) Efficacy score (c) Paraphrase score (d) Neighborhood score

Figure 3: Performance of FastEMMET in Llama 2 across different batch sizes



(a) Overall Score (b) Efficacy score (c) Paraphrase score (d) Neighborhood score

Figure 4: Performance of FastMEMIT in Llama 2 across different batch sizes

in equation 2 (Gupta et al., 2025). Note that this is needed only for batch sizes less than 10. With $d_m = 10$, the editing performance for Llama2-7B is reliably close to the full precomputation performance for both algorithms. This requires approximately 0.25% tokens when compared to the full pre-computation.

6 Related Work

Knowledge editing methods can broadly be divided into two categories - in-context editing and parameter-modifying methods. In-context editing techniques, such as SERAC (Mitchell et al., 2022), ICE (Cohen et al., 2023), MeLLo (Zhong et al., 2023) and GRACE (Hartvigsen et al., 2023), allow updated knowledge to be added temporarily by providing new information in the model context. On the other hand, parameter-modifying knowledge editing do this by infusing new knowledge in the model weights. MEMIT (Meng et al., 2022b) and ROME (Meng et al., 2022a) are two notable methods in this area that offer efficient solutions to directly edit the model parameters and are closely related to model interpretability. ROME introduced the idea of identifying key layers that store factual knowledge and then updating the corresponding weights to edit the model. MEMIT extended this approach by enabling batched editing, allowing multiple facts to be edited at once using a closed-form solution. These methods have been very popular and have seen a growing body of work in recent times that overcome various limitations at scale (Gupta et al., 2024b). These include methods like PMET (Li et al., 2023), EMMET (Gupta et al., 2024c), PRUNE (Ma et al., 2024), AlphaEdit (Fang et al., 2024).

7 Conclusion

In this paper, we significantly reduce the upfront precomputation time required to cache hidden representation of a model before editing can begin for locate-then-edit methods like MEMIT, ROME and EMMET. We do this by first finding the theoretical minimum number of precomputation tokens required. We then empirically search for the optimal 'minimum' number of precomputation tokens required to perform successful editing without compromising performance. **Our recommendation is to use 10 times the theoretical minimum of tokens, or to use a dynamic multiplier of 10**. Note that this number is less than 0.4% of the originally used 44 million tokens. However, this number can further be reduced for specific models and editing algorithms as shown in our paper. This study allows editing for a new model to beging within a few minutes, saving many hours of precomputation time.

8 Limitations

In our work, we present optimal number of tokens required for precomputation for popular knowledge editing methods. We evaluate this in the setting

of singular and batched editing. A recently popular mode of editing is sequential editing (Fang et al., 2024). We leave evaluation of optimal precomputation requirements for sequential editing to future work. Additionally, it has been shown that sequential editing also leads to loss of downstream performance (Gupta et al., 2024b). In this work, we do not analyze the relationship between the number of precomputation tokens and downstream performance, which we also leave for future work.

References

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2023. Evaluating the ripple effects of knowledge editing in language models. *arXiv preprint arXiv:2307.12976*.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Xiang Wang, Xiangnan He, and Tat-seng Chua. 2024. Alphaedit: Null-space constrained knowledge editing for language models. *arXiv preprint arXiv:2410.02355*.

Akshat Gupta, Sidharth Baskaran, and Gopala Anumanchipalli. 2024a. Rebuilding rome: Resolving model collapse during sequential model editing. *arXiv preprint arXiv:2403.07175*.

Akshat Gupta, Phudish Prateepamornkul, Maochuan Lu, Ahmed Alaa, Thomas Hartvigsen, and Gopala Anumanchipalli. 2025. Lifelong knowledge editing requires better regularization. *Preprint*, arXiv:2502.01636.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. 2024b. Model editing at scale leads to gradual and catastrophic forgetting. *arXiv preprint arXiv:2401.07453*.

Akshat Gupta, Dev Sajnani, and Gopala Anumanchipalli. 2024c. A unified framework for model editing. *arXiv preprint arXiv:2403.14236*.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in Neural Information Processing Systems*, 36:47934–47959.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2023. Pmet: Precise model editing in a transformer. *arXiv preprint arXiv:2308.08742*.

Jun-Yu Ma, Hong Wang, Hao-Xiang Xu, Zhen-Hua Ling, and Jia-Chen Gu. 2024. Perturbation-restrained sequential model editing. *arXiv preprint arXiv:2405.16821*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models, 2023. *URL https://arxiv.org/abs/2307.09288*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*.

Junsang Yoon, Akshat Gupta, and Gopala Anumanchipalli. 2024. Is bigger edit batch size always better?–an empirical study on model editing with llama-3. *arXiv preprint arXiv:2405.00664*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*.

A Appendix

Batch Size	Num Batches	Total Edits
1	1000	1000
16	10	160
64	5	320
256	5	1280
1024	3	3072

Table 1: Statistics for batch size and number of batches used to create the numbers for this paper.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.9	50.4	94.0	50.35	65.59	49.09	83.57	49.93
16	100.0	70.62	96.25	60.31	74.19	47.69	88.57	58.01
64	100.0	90.31	95.94	72.03	73.56	54.37	88.18	69.20
256	99.84	91.95	96.13	70.23	67.14	54.83	84.95	69.20
1024	99.32	86.72	92.15	91.62	68.28	54.11	81.35	67.17

Table 2: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 1 in GPT2-XL on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.9	100.0	94.0	94.25	65.59	63.61	83.57	82.57
16	100.0	100.0	96.25	96.88	74.19	72.44	88.57	87.90
64	100.0	99.69	95.94	96.88	73.56	71.03	88.18	87.12
256	99.84	99.84	96.13	94.34	67.14	64.26	84.95	82.92
1024	99.32	98.89	92.15	89.86	68.28	59.23	81.35	78.69

Table 3: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 2 in GPT2-XL on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.9	100.0	94.0	94.35	65.59	64.91	83.57	83.32
16	100.0	100.0	96.25	96.25	74.19	72.44	88.57	87.73
64	100.0	99.69	95.94	97.19	73.56	72.28	88.18	87.83
256	99.84	99.69	96.13	94.14	67.14	65.72	84.95	83.63
1024	99.32	99.06	92.15	90.97	68.28	60.79	81.35	79.91

Table 4: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 3 in GPT2-XL on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.9	100.0	94.0	94.2	65.59	64.95	83.57	83.30
16	100.0	100.0	96.25	95.94	74.19	74.31	88.57	88.54
64	100.0	99.69	95.94	96.56	73.56	72.10	88.18	87.61
256	99.84	99.92	96.13	95.86	67.14	66.18	84.95	84.38
1024	99.32	99.19	92.15	91.62	62.68	61.18	81.35	80.33

Table 5: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 4 in GPT2-XL on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.9	100.0	94.0	93.25	65.59	66.55	83.57	83.89
16	100.0	99.38	96.25	95.94	74.19	74.38	88.57	88.41
64	100.0	99.69	95.94	95.0	73.56	73.38	88.18	87.75
256	99.84	99.69	96.13	95.55	67.14	66.4	84.95	84.37
1024	99.38	99.19	92.15	92.01	62.68	61.88	81.35	80.88

Table 6: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 10 in GPT2-XL on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	97.2	50.6	86.55	51.25	71.14	48.28	83.56	50.01
16	96.25	51.88	82.5	44.69	78.19	53.0	84.98	49.57
64	97.19	83.12	86.41	67.97	77.56	60.09	86.31	69.14
256	96.88	85.23	86.48	56.17	72.86	68.52	84.24	67.98
1024	95.48	91.6	84.94	76.07	70.38	59.37	82.29	73.33

Table 7: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 1 in GPT2-XL on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	97.2	100.0	86.55	92.45	71.14	67.12	83.56	83.99
16	96.25	100.0	82.5	92.19	78.19	76.31	84.98	88.36
64	97.19	100.0	86.41	95.16	77.56	75.47	86.31	88.86
256	96.88	99.77	86.48	94.3	72.86	69.94	84.24	85.89
1024	95.48	99.38	84.94	92.24	70.38	67.61	82.29	84.04

Table 8: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 2 in GPT2-XL on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	97.2	99.6	86.55	91.1	71.14	71.01	83.56	85.47
16	96.25	100.0	82.5	90.00	78.19	77.50	84.98	88.20
64	97.19	98.44	86.41	91.09	77.56	76.91	86.31	87.88
256	96.88	99.06	86.48	92.70	72.86	71.48	84.24	86.03
1024	95.48	98.76	84.94	90.92	70.38	68.43	82.29	83.94

Table 9: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 3 in GPT2-XL on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	97.2	99.4	86.55	91.3	71.14	71.54	83.56	85.73
16	96.25	98.75	82.5	86.56	78.19	77.75	84.98	86.85
64	97.19	98.44	86.41	90.16	77.56	77.12	86.31	87.67
256	96.88	99.06	86.48	90.43	72.86	71.73	84.24	85.48
1024	95.48	98.37	84.94	89.01	70.38	69.29	82.29	83.72

Table 10: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 4 in GPT2-XL on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	97.2	98.4	86.55	90.2	71.14	72.84	83.56	85.76
16	96.25	96.25	82.5	84.38	78.19	78.06	84.98	85.58
64	97.19	97.19	86.41	87.66	77.56	77.47	86.31	86.69
256	96.88	97.89	86.48	88.48	72.86	72.81	84.24	85.10
1024	95.48	97.04	84.94	87.65	70.38	69.85	82.29	83.26

Table 11: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 10 in GPT2-XL on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.9	51.6	94.95	49.85	77.59	50.01	89.73	50.47
16	100.0	60.0	93.44	53.44	81.25	46.69	90.88	52.81
64	99.69	58.75	93.91	52.5	81.78	52.59	91.16	54.46
256	99.45	98.12	94.14	86.6	78.62	54.66	89.82	86.6
1024	99.67	95.15	93.67	82.62	74.27	54.98	87.78	73.52

Table 12: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 1 in GPT-J on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.9	100.0	94.95	97.4	77.59	71.81	89.73	87.73
16	100.0	100.0	93.44	96.56	81.25	79.88	90.88	91.25
64	99.69	100.0	93.91	96.88	81.78	80.03	91.16	91.41
256	99.45	99.84	94.14	97.19	78.62	74.18	89.82	88.79
1024	99.67	99.8	93.67	96.35	74.27	68.01	87.78	85.46

Table 13: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 2 in GPT-J on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.9	100.0	94.95	96.65	77.59	75.2	89.73	89.16
16	100.0	100.0	93.44	94.38	81.25	80.88	90.88	91.02
64	99.69	100.0	93.91	96.56	81.78	80.72	91.16	91.61
256	99.45	99.84	94.14	96.99	78.62	75.73	89.82	89.46
1024	99.67	99.8	93.67	96.14	74.27	70.16	87.78	86.51

Table 14: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 3 in GPT-J on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.9	100.0	94.95	96.85	77.59	74.72	89.73	88.99
16	100.0	100.0	93.44	95.0	81.25	81.12	90.88	91.31
64	99.69	100.0	93.91	94.06	81.78	81.72	91.16	91.27
256	99.45	99.92	94.14	96.33	78.62	76.45	89.82	89.63
1024	99.67	99.74	93.67	95.8	74.27	71.03	87.78	86.84

Table 15: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 4 in GPT-J on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.9	100.0	94.95	95.7	77.59	76.55	89.73	89.51
16	100.0	100.0	93.44	94.06	81.25	81.19	90.88	91.05
64	99.69	100.0	93.91	94.53	81.78	81.78	91.16	91.44
256	99.45	99.77	94.14	94.8	78.62	77.27	89.82	89.51
1024	99.67	99.64	93.67	94.91	74.27	73.24	87.78	87.65

Table 16: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 10 in GPT-J on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	100.0	51.3	94.75	49.8	80.34	50.12	86.27	50.39
16	100.0	43.12	96.56	52.55	80.19	46.88	91.38	47.19
64	100.0	46.88	96.09	47.97	81.28	52.56	91.71	49.01
256	99.77	75.55	96.02	62.23	78.02	55.02	90.21	63.18
1024	99.74	76.4	94.66	65.27	75.65	56.43	88.73	65.03

Table 17: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 1 in GPT-J on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	100.0	100.0	94.75	96.9	80.34	67.79	86.27	85.53
16	100.0	100.0	96.56	96.88	80.19	80.38	91.38	91.56
64	100.0	100.0	96.09	96.41	81.28	80.41	91.71	91.43
256	99.77	99.77	96.02	96.88	78.02	77.16	90.21	90.07
1024	99.74	99.71	94.66	95.49	75.65	74.05	88.73	88.22

Table 18: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 2 in GPT-J on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	100.0	100.0	94.75	96.75	80.34	72.72	86.27	88.00
16	100.0	100.0	96.56	97.19	80.19	80.62	91.38	91.76
64	100.0	100.0	96.09	97.03	81.28	81.09	91.71	91.91
256	99.77	99.84	96.02	96.56	78.02	77.77	90.21	90.27
1024	99.74	99.71	94.66	95.48	75.65	74.88	88.73	88.60

Table 19: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 3 in GPT-J on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	100.0	100.0	94.75	96.45	80.34	73.78	86.27	88.43
16	100.0	100.0	96.56	96.56	80.19	80.62	91.38	91.57
64	100.0	100.0	96.09	96.88	81.28	80.53	91.71	91.63
256	99.77	99.84	96.02	96.52	78.02	78.05	90.21	90.39
1024	99.74	99.74	94.66	95.12	75.65	75.65	88.73	88.89

Table 20: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 4 in GPT-J on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	100.0	100.0	94.75	95.65	80.34	74.85	86.27	88.71
16	100.0	100.0	96.56	96.25	80.19	80.75	91.38	91.53
64	100.0	100.0	96.09	96.72	81.28	81.31	91.71	91.91
256	99.77	99.69	96.02	95.43	78.02	78.38	90.21	90.17
1024	99.74	99.71	94.66	94.61	75.65	76.12	88.73	88.92

Table 21: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 10 in GPT-J on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.5	51.2	98.5	49.45	59.0	50.15	80.75	50.26
16	99.38	52.5	95.62	49.69	82.94	51.56	92.08	51.22
64	98.44	49.38	97.19	47.66	78.0	52.03	90.17	49.62
256	99.61	75.31	97.89	60.86	62.1	51.25	82.51	60.94
1024	98.73	73.44	96.14	62.65	57.17	50.34	78.90	60.67

Table 22: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 1 in Llama 2 on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.5	99.8	98.5	99.0	59.0	49.54	80.75	74.42
16	99.38	99.38	95.62	98.12	82.94	72.25	92.08	87.98
64	98.44	100.0	97.19	97.81	78.0	66.09	90.17	84.85
256	99.61	98.98	97.89	95.74	62.1	53.37	82.51	76.36
1024	98.73	94.66	96.14	88.64	57.17	51.49	87.78	78.9

Table 23: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 2 in Llama 2 on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.5	99.9	98.5	99.0	59.0	52.04	80.75	76.28
16	99.38	98.12	95.62	96.25	82.94	77.19	92.08	89.45
64	98.44	99.69	97.19	98.44	78.0	70.94	90.17	87.49
256	99.61	99.3	97.89	97.62	62.1	56.05	82.51	78.62
1024	98.73	97.59	96.14	92.74	57.17	52.24	78.90	74.67

Table 24: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 3 in Llama 2 on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.5	99.7	98.5	98.0	59.0	54.39	80.75	77.68
16	99.38	100.0	95.62	97.81	82.94	79.25	92.08	91.34
64	98.44	99.69	97.19	97.81	78.0	71.53	90.17	87.62
256	99.61	99.22	97.89	96.72	62.1	57.05	82.51	79.05
1024	98.73	97.85	96.14	94.32	57.17	52.86	78.90	75.49

Table 25: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 4 in Llama 2 on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET	EMMET	FastEMMET
1	99.5	99.8	98.5	98.25	59.0	56.72	80.75	79.30
16	99.38	99.38	95.62	96.88	82.94	81.81	92.08	92.00
64	98.44	99.69	97.19	97.5	78.0	75.53	90.17	89.47
256	99.61	99.22	97.89	98.2	62.1	59.17	82.51	80.72
1024	98.73	98.37	96.14	95.88	57.17	54.74	78.90	77.19

Table 26: Comparison between EMMET and FastEMMET for multiple layers with different batch sizes, dynamic multiplier = 10 in Llama 2 on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	96.6	53.9	89.4	54.45	60.82	51.81	78.98	53.36
16	99.38	46.25	99.38	44.38	65.12	51.44	84.55	47.17
64	98.12	49.69	97.03	48.59	61.09	50.94	81.37	49.72
256	96.33	79.06	93.59	65.39	56.85	50.9	77.60	63.04
1024	93.95	65.89	90.45	55.37	60.17	49.98	78.28	56.34

Table 27: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 1 in Llama 2 on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	96.6	72.5	89.4	72.5	60.82	53.81	78.98	64.97
16	99.38	97.5	99.38	97.5	65.12	59.81	84.55	80.57
64	98.12	95.31	97.03	95.0	61.09	58.66	81.37	78.81
256	96.33	89.14	93.59	81.68	56.85	54.0	77.60	71.46
1024	93.95	88.57	90.45	78.04	60.17	54.11	78.28	70.44

Table 28: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 2 in Llama 2 on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	96.6	90.0	89.4	85.95	60.82	55.36	78.98	73.51
16	99.38	99.38	99.38	97.5	65.12	67.56	84.55	85.42
64	98.12	98.12	97.03	96.56	61.09	62.13	81.37	81.87
256	96.33	95.08	93.59	91.05	56.85	55.72	77.60	76.05
1024	93.95	90.69	90.45	82.75	60.17	55.51	78.28	72.94

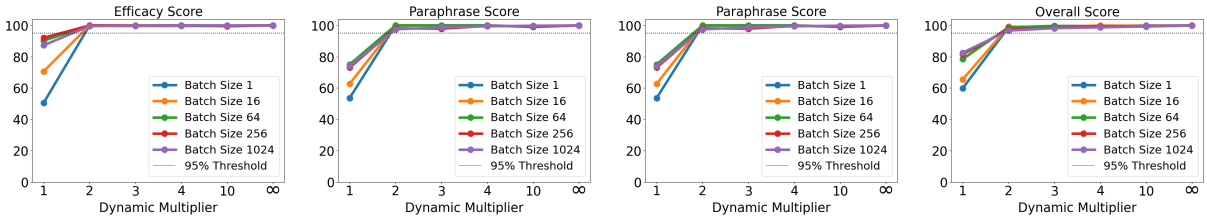
Table 29: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 3 in Llama 2 on the CounterFact dataset.

Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	96.6	94.9	89.4	91.05	60.82	57.15	78.98	76.88
16	99.38	98.75	99.38	98.44	65.12	70.19	84.55	86.87
64	98.12	98.12	97.03	96.09	61.09	63.06	81.37	82.29
256	96.33	95.08	93.59	90.51	56.85	57.33	77.60	76.90
1024	93.95	93.42	90.45	90.22	60.17	59.34	78.28	77.63

Table 30: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 4 in Llama 2 on the CounterFact dataset.

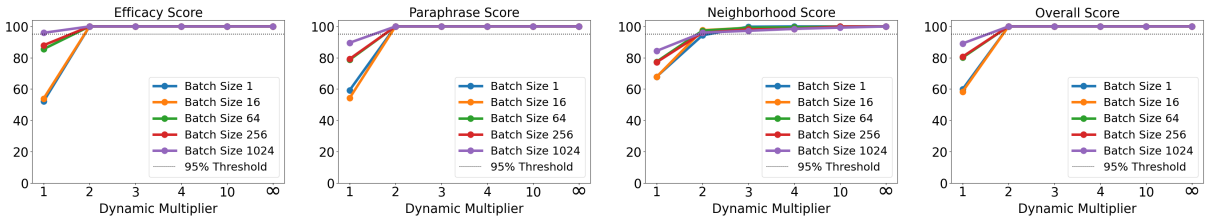
Batch Size	ES (Efficacy)		PS (Generalization)		NS (Locality)		S (Score)	
	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT	MEMIT	FastMEMIT
1	96.6	98.7	89.4	93.05	60.82	67.07	78.98	83.82
16	99.38	98.75	99.38	98.75	65.12	75.37	84.55	89.49
64	98.12	98.44	97.03	97.5	61.09	69.88	81.37	86.39
256	96.33	97.42	93.59	94.26	56.85	57.93	77.60	78.66
1024	93.95	93.42	90.45	90.22	60.17	59.34	78.28	77.63

Table 31: Comparison between MEMIT and FastMEMIT for multiple layers with different batch sizes, dynamic multiplier = 10 in Llama 2 on the CounterFact dataset.



(a) Efficacy score (b) Paraphrase score (c) Neighborhood score (d) Overall Score

Figure 5: Performance of FastEMMET in GPT2-XL across different batch sizes



(a) Efficacy score (b) Paraphrase score (c) Neighborhood score (d) Overall Score

Figure 6: Performance of FastMEMIT in GPT2-XL across different batch sizes

Meaning Variation and Data Quality in the Corpus of Founding Era American English

Dallas Card
University of Michigan School of Information*
dalc@umich.edu

Abstract

Legal scholars are increasingly using corpus based methods for assessing historical meaning. Among work focused on the so-called founding era (mid to late 18th century), the majority of such studies use the Corpus of Founding Era American English (COFEA) and rely on methods such as word counting and manual coding. Here, we demonstrate what can be inferred about meaning change and variation using more advanced NLP methods, focusing on terms in the U.S. Constitution. We also carry out a data quality assessment of COFEA, pointing out issues with OCR quality and metadata, compare diachronic change to synchronic variation, and discuss limitations when using NLP methods for studying historical meaning.

1 Introduction

Alongside dictionaries and other reference material, legal scholars are increasingly turning towards historical text corpora in order to make arguments about the historical meanings of terms, in so far as they are relevant to modern legal questions. While much of this work involves manually inspecting usages of individual terms, here we demonstrate the use of more recent NLP methods for assessing meaning change and variation, relying on masked language models (MLMs) and focusing on terms in the U.S. Constitution.

The main corpus used in investigating historical legal meanings from the so-called founding era (roughly the second half of the 18th century), is the Corpus of Founding Era American English (COFEA; Hashimoto, 2023), which brings together a series of text collections, including broadsides, legal statutes, debates, and letters. Because COFEA has become so central to arguments about legal meaning, it is essential to investigate both the limitations of the corpus, and what we can learn from it with using state of the art NLP techniques. Using a full-text copy of the corpus, we first investigate and describe its contents, noting issues with unreliable metadata and the quality of the optical character recognition (OCR). In contrast to the in-depth study of individual words, as is more commonly done, we then present a broader automated investigation of meaning, along with an extensive discussion of the limitations of this approach.

The structure of this paper is as follows: we first describe and investigate COFEA (§3), noting several issues and limitations, especially with respect to the OCR and metadata (§4). Using MLMs, we then measure change and variation in the meaning of terms in the U.S. Constitution (§5), both across time and across collections, finding the former to be more extensive than the latter. Finally, we consider the inferred meanings of individual terms in the specific context of the Constitution, finding suggestive evidence of an overall bias towards more formal, as opposed to popular, meanings (§5.3), and end with a discussion of limitations. Data processing and analysis code, as well as interactive versions of all figures, are available online.[1]

2 Background

The U.S. Constitution was drafted in the summer of 1787, and quickly embraced as the fundamental law of the United States. As described in Gienapp (2018), disagreements began almost immediately as to how it should be interpreted, and continue to this day. In recent decades, much of this debate has been with respect to the theory of *originalism*, which is broadly the idea that the original meaning of the Constitution should remain in force.[2]

Independent of fundamental legal theories and questions, most scholars recognize the importance and difficulty of knowing and characterizing the

*Part of this work done while at Stanford University.

[1] https://dallascard.github.io/cofea

[2] There is an enormous literature on originalism which we do not attempt to survey here, but note there are a range of positions which all go under this name; see Solum (2019).

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 841–856
July 27 - August 1, 2025

meaning of language. Among other tools used by legal scholars for this purpose is that of *corpus linguistics*—using a corpus of texts to attempt to understand the meaning of particular terms in context (Mouritsen, 2010). Since COFEA was introduced, in part to help answer questions about "ordinary meaning" in the founding era (Hashimoto, 2023), it has quickly become the main reference corpus for this type of investigation.[3]

Indeed, a popular genre of legal scholarship focuses on the meanings of specific terms or phrases, such as "citizens" (Stout et al., 2020), "carry" (Mouritsen, 2010), or "bear arms" (Goldfarb, 2019). Most of these papers tend to use relatively straightforward methods, such as colocations, keyword-in-context, or manual annotations (e.g,. Barclay et al., 2019; Stout et al., 2020), but increasingly we are seeing legal scholars experiment with more advanced NLP techniques (e.g., Nyarko and Sanga, 2022; Livermore et al., 2024).

In this paper, we draw particular inspiration from Lee and Phillips (2019), who focus on handful of terms, like *commerce* and *domestic violence*, using collocations and manual coding, to examine both how the meanings of these terms have changed since the founding era, and the extent of variation within COFEA. Specifically, we leverage the capability of MLMs to assess meaning, considering all terms in the Constitution. While remaining agnostic with respect to debates about originalism, our purpose is both to illustrate the power of such methods, but also to demonstrate and discuss the limitations that these data and methods entail.

3 Data and Preprocessing

We first acquire a full text copy of COFEA from its creators. Although the corpus is made up of six collections, the vast majority of words come from three of these: books, pamphlets, and broadsides (EVANS), letters of the U.S. "Founding Fathers" (FOUNDERS), and a collection of mostly legal documents provided by HeinOnline (HEIN). Smaller collections include laws and resolutions enacted by Congress (STATUTES), records of the 1787 Constitutional Convention (FARRANDS), and a collection of the corresponding debates in state conventions (ELLIOTS). To narrow our focus, we limit our analysis to the period from 1760–1800, which is the period of greatest overlap across sources,

[3]COFEA is publicly searchable via a web interface at https://lawcorpus.byu.edu/cofea.

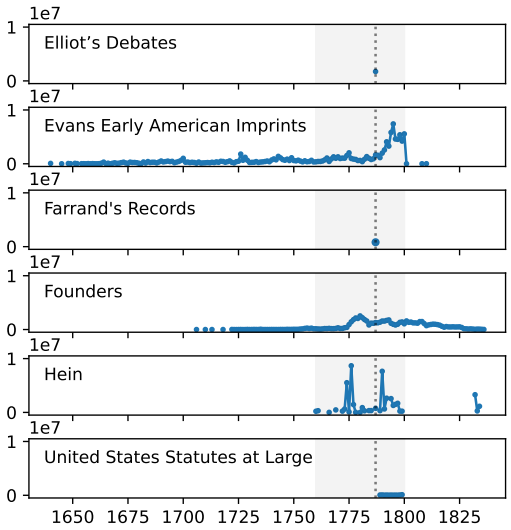


Figure 1: Number of tokens per year in the six collections that comprise COFEA. Grey bands indicate the period studied in this paper, and the dotted line shows the year in which the U.S. Constitution was written. Note that years associated with HEIN documents may in some cases by unreliable (see §4).

as shown in Figure 1. For additional information about COFEA, please refer to Appendix A.

To augment COFEA, we also include all newspaper articles from those years published in *The Pennsylvania Gazette* (TPG), which has previously been used for investigating historical legal language (Strang, 2018). For a modern reference corpus, we use the Corpus of Contemporary American English (COCA; Davies, 2008). For the U.S. Constitution, we use the text provided by the National Archives, including the Bill of Rights. After some initial filtering and preprocessing of COFEA, we tokenize all documents using BERT (Devlin et al., 2019), do spelling normalization, and curate a list of bigrams based on normalized pointwise mutual information, which we augment with the bigrams from Lee and Phillips (2019). Additional details on sources and preprocessing are given in Appendices A and B.

4 Data Quality Assessment

Before investigating meaning change, we first assess the quality of the source documents in COFEA. During initial filtering, we note the presence of a number of documents that contain modern text, such as editorial notes, which we exclude (see Appendix B.1). In addition, the years associated with some documents seem to be incorrect or imprecise. This is especially the case for documents in HEIN, some of which collect together

texts written over multiple years. For our purposes, this makes little difference, as most texts in HEIN appear to date from roughly 1760–1800. However, this could be a more serious issue for investigations which consider precise timing, such as before and after the Constitution was written, or documents from a particular year (see Appendix B.2).

To measure the quality of the OCR for each corpus, we use a simple but commonly used technique of checking for coverage using a dictionary (Springmann et al., 2014; Neudecker et al., 2021).[4] After lemmatizing all documents, we measure the proportion of lemmas that appear in the Webster's 1913 dictionary (Porter, 1913), enhanced with additional terms such as titles, names, places, and abbreviations (see Appendix C.1 for details).[5]

Figure 2 shows a plot of the OCR quality of each corpus based on this measure, where each dot represents one document, sorted by score, and evenly distributed across the x-axis, on a log scale. As can be seen, some of the worst documents are in EVANS and FOUNDERS, but overall HEIN has the worst quality over most of the range. Inspecting the data, the most common tokens in HEIN that are not in the dictionary appear to be poor OCR renderings of terms like "shall" (e.g., "thall") and "justice" (e.g., "juflice"), as well as subwords from improper splitting, like "tion". This means that many occurrences of some terms in the HEIN collection will effectively be missed by a simple keyword search, which is the primary way that scholars access COFEA. For alternative methods of assessing OCR quality, with broadly similar results, please refer to Appendix C.

5 Change and Variation in Meaning

5.1 Measuring changes in meaning

To investigate both change and variation in word meaning, we make use of masked language models, specifically `bert-large-uncased`. In particular, we borrow the technique from Card (2023), which uses changes in the most probable substitutes for terms as informative of word meaning, as this approach allows for both aggregate comparisons,

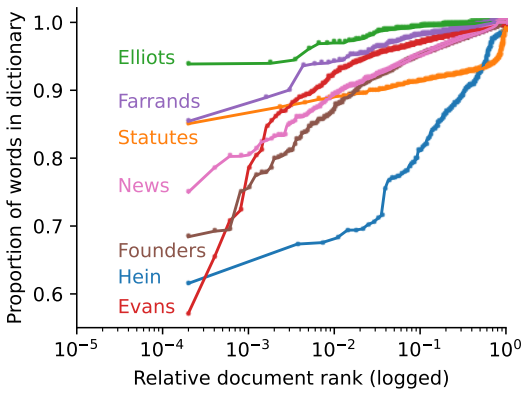


Figure 2: OCR quality across corpora as measured by coverage in the augmented Webster's 1913 dictionary. Each point represents one document, showing the proportion of words in that document found in the dictionary, ranked and distributed across the x-axis.

and inspecting individual term mentions.[6]

For all terms that occur in the Constitution, we sample up to 4000 instances from each of COFEA and COCA, along with samples of additional random words, and pass each word, masked and in context, through the model. Following Eyal et al. (2022), we save the top-k most probable substitutes (with $k = 10$), excluding stopwords (details in Appendix D). For measuring changes in word meaning, we take the total counts of the top-k substitutes for all instances of the word in a corpus, to get a distribution over the vocabulary, and then compute the Jensen-Shannon divergence (JSD) between the distributions for each corpus.

5.2 Diachronic Changes in Meaning

Using the method described above, we first compare COFEA documents from 1760–1800 to modern documents COCA. In doing so, we validate that there have indeed been considerable changes in meaning. As illustrative examples, the terms with the largest measured change in meaning over time are listed in Table 1, along with common probable substitutes from the model. We note that some of those identified involve different grammatical senses (e.g., *captures* as a noun rather than a verb), whereas others represent different meanings (e.g., *quartered* as stationed rather than divided).

Among constitutional terms that have changed the most, many of these had historical military meanings (e.g., *captures*, *quartered*, *training*, *ar-*

[4] According to Hashimoto (2023), documents in COFEA are being evaluated for OCR accuracy, but these scores are not yet available, to the best of our knowledge.

[5] Although this dictionary is from more than a century after most of the COFEA documents, dictionaries are known to lag behind linguistic change, and preserve some historical meanings, making this a reasonable choice.

[6] For comparison to a simpler method of measuring change in meaning, we also include a parallel analysis using the approach of Hamilton et al. (2016) in Appendix D.

Term	Founding era	Modern era	JSD
captures	captures prizes capture seizures prize	captures reflects shows represents describes	0.91
domestic violence	invasion insurrection violence invasions	violence abuse rape crime assault	0.91
marque	mar truce protection war commission	porte salle junta grange crescent	0.90
capitation	direct poll land general state	medicare payment insurance compensation	0.90
quartered	stationed posted kept placed lodged	sliced chopped peeled seeded trimmed	0.86
affirmation	declaration oath certificate deposition	expression recognition acceptance assertion	0.86
training	training raising bringing exercising trained	training education instruction practice	0.85
piracies	crimes offenses piracy murders treason	crimes crime abuses offenses acts	0.85

Table 1: Constitutional terms with the largest meaning change from the founding to modern era, with most common substitutes, and corresponding JSD values. For an expanded list, see Table 7 in the Appendix.

senals). In addition, compared to a random set of background terms, we find that the constitutional terms have changed slightly but significantly more than other terms since the founding era, even after correcting for frequency (see analysis in Appendix D). As a sanity check, we also find that the terms showing the *least* change in meaning tend to be numbers and names of months or days, as was previously exploited by past work (Nyarko and Sanga, 2022). In this case, all but one of the top 20 lowest JSD scores correspond to numbers or temporal terms (e.g., *days*, *years*).

5.3 Specialized vs. Popular Meanings

An enduring debate about the Constitution is the extent to which it was written using a specialized legal vocabulary, as opposed to an accessible popular vernacular (McGinnis and Rappaport, 2018; Gienapp, 2018). Looking first at word counts, we find that, on average, constitutional terms appear more frequently in legal corpora than others. Figure 3 shows the relative frequencies of all terms in the Constitution in Popular, Legal, and Founders documents, projected onto the simplex.[7] As can be seen, most of the terms in the Constitution are relatively evenly represented across these three types of sources. However, on average, there are more terms that appear more commonly in legal sources, rather than popular sources or Founders' papers.

Frequency alone, however, does necessarily tell us about meaning. To that end, we carry out a brief comparison to provide suggestive evidence on this question. To do so, we use the same technique as above for studying change in meaning, but here apply it to variation across parts of the COFEA corpus representing legal vs. popular documents.

Overall, we find the synchronic variation between legal and popular sources is much less than

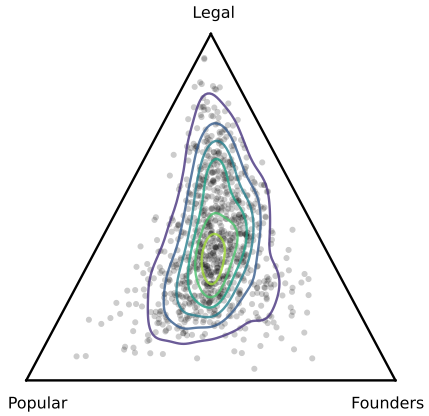


Figure 3: Overall, constitutional terms are more common in legal than other sources. Each point represents one term, with relative frequencies across the three types of sources projected on to the simplex, with contours from kernel density estimation overlaid on top.

the diachronic change in meaning from the founding to the modern era: the mean JSD across all constitutional terms is 0.45 for the former and 0.62 for the latter. Nevertheless, there are many terms which show clear differences in terms of primary meaning across sources. The terms with the largest differences are shown in Table 2, excluding those due to obvious OCR errors.

The above analysis tells us about the most common meanings, but in principle, MLMs also allow us to investigate the specific usage of terms in the Constitution itself. To get at this, we repeat the process described above to embed each specific token in the Constitution, and compare the top ten replacements suggested for the constitutional context to the most common replacements from each of the legal and popular subsets. For each mention, we count the token as leaning towards a specialized (legal) vs. popular usage if the set overlap is at least two greater for one than the other; otherwise, we count the mention as indeterminate. For words that

[7] For legal documents, we use all of STATUTES, FARRANDS, and ELLIOTS, and the parts of HEIN marked as "Legal". For popular sources, we use EVANS and TPG.

Term	Legal	Popular	JSD
tender	tender payment demand currency money	tender kind soft generous great	0.80
dock	dock ship navy naval docks	dock market water street front	0.75
bankruptcies	commerce trade religion slaves debts	losses debts commerce trade failures	0.71
affecting	affecting respecting touching concerning	affecting interesting awful melancholy	0.70
repassed	rejected amended approved repealed	passed crossed entered left ascended	0.70
resignation	resignation removal refusal appointment	resignation submission patience obedience	0.70
sign	sign receive make deliver take	sign head tavern foot signs	0.69
searches	search searches seizures arrests attacks	searches search knows sees inquiries	0.69

Table 2: Constitutional terms with the largest difference in meaning between legal and popular sources in COFEA, (excluding those due to obvious OCR errors), shown with most common substitutes, and corresponding JSD values. All terms here lean towards typical legal meanings when used in the Constitution, according to the analysis described in section §5.3. See Table 10 in the Appendix for an expanded list.

occur more than once in the Constitution, we take the majority outcome, defaulting to indeterminate in the case of ties (see Appendix D.3 for details).

Inspecting the top 40 words with greatest difference between sources, we find only two that lean towards a popular meaning, and in both cases (*aid* and *fix*), these can be attributed to OCR errors. Of the remainder, 26 lean towards a specialized legal meaning, and the remaining 12 are indeterminate. Thus, for the terms in the Constitution that show the greatest variation in meaning across sources, most of these align more with how the terms were used in legal documents, and almost none with usage in more popular sources.

6 Discussion

Despite the popularity of corpus linguistics for assessing historical legal meanings, numerous criticisms have been directed at this approach in general, and COFEA in particular. With respect to COFEA itself, a major concern has to do with it being a non-representative sample of text. In particular, a vastly disproportionate amount of text in COFEA comes from a handful of people (the so-called "Founding Fathers"). As such, COFEA as a whole over-represents elite voices, and under-represents more popular sources.

With respect to the use of corpus linguistics in law, additional critiques have focused on biases in the sources included (Drakeman, 2020; Jennejohn et al., 2021), subjectivity in interpretation (Gries, 2020), placing too much weight on frequent usages (Herenstein, 2017), the historical or linguistic competence required in interpretation (Slocum and Gries, 2020), and lack of clear and reproducible methodology (Henderson et al., 2024). While our more computational approach has the advantage of being fully reproducible, other criticisms still apply, especially with respect to source bias.

Beyond the reasons for caution noted by others, we additionally demonstrate the presence of some modern text in COFEA, as well as some OCR errors and potentially unreliable dates, especially in HEIN. In addition, it is important to note that the methods we have used here primarily establish a single, relatively coarse sense for each time period or collection, and may miss more subtle nuances of meaning. While other methods could potentially exhibit greater sensitivity to such nuances, there is inherently only so much information in a single occurrence of a word. We can thus have much more confidence in aggregate comparisons, even though they lack some sensitivity to context.

Nevertheless, without implying any particular legal relevance, we can say that the extent of differences between specialized vs. popular meanings in the founding era is small compared to meaning changes over time, and that the constitutional terms that differ in meaning across source types seem to broadly lean towards legal senses. Confident judgements about any specific terms would require more input from legal and historical scholars, but overall our results highlight the importance of combining careful manual inspection with reproducible computational analyses on known data.

7 Conclusion

This paper provides the first external assessment of data quality in COFEA, and the first computational analysis of meaning change and variation for all terms in the U.S. Constitution. While noting limitations with respect to both methods and data, especially issues involving dates and OCR quality for the HEIN collection, we find that variation in meaning across sources is much less than change in meaning over time, with weak evidence in favor of the broadly specialized (legal) character of the language in the U.S. Constitution.

Limitations

Although the use of more sophisticated techniques than word counting or manual coding can help to shed some light on historical meanings, we should nevertheless note considerable limitations with the analyses presented here.

First, building on the limitations noted in the Discussion, COFEA itself is far from representative of all text produced during the founding era. Although EVANS claims to be close to comprehensive of certain media (broadsides, books, and pamphlets), even the language used in these is still no doubt far from how people of the time typically spoke in conversation. Moreover, COFEA as a whole is missing newspaper articles, which were an important venue for public communication. Here, we have augmented COFEA with *The Pennsylvania Gazette* because it is easily accessibility, but other sources would be useful to consider.

In terms of the methods used here, our approach provides one dominant sense per time period (or per source), when in fact many terms may exhibit multiple meanings within any collection of documents. As such, our analyses are primarily getting at the most common meanings, and may miss less common senses. It is entirely possible, for example, that the more common sense of a term in the legal sources was also used in popular sources, but with less frequency than another sense. Although one could try to inspect individual usages in each corpus, inferences based on single usages will be much noisier than collection averages.

In addition, the meanings inferred via our approach are relatively coarse. For example, a notable legal case hinged on whether "carrying" a weapon should include transporting it in a vehicle (Mouritsen, 2010). Looking only at common replacements for *carry* in COFEA can help us to get a general sense of its meaning (e.g., bring, take, put), but this may tell us relatively little about the subtleties of implied meaning. As such, this approach is unlikely to be an adequate substitute for historical analysis; at best, computational methods can complement other approaches.

To compare with past work, we opted to use some common phrases, but how to best identify bigrams or other multiword expressions is open to debate. Although the method we use gives reasonable results, we note that the bigrams we included from Lee and Phillips (2019) are quite far from meeting our criteria. This raises the question of whether such phrases (e.g., *public use*) are best understood as distinct concepts or not, and illustrates the importance of having some criteria for deciding what to include. Because meaning is partly but not entirely compositional, studying the meaning of a particular word as opposed to a phrase across all mentions could lead to different results.

Regarding what can be inferred, it is also worth noting that the evidence for certain terms is extremely limited. For example, there are fewer than 100 mentions of the phrase *domestic violence* in COFEA. Of these, only a handful appear in documents that were published before the U.S. Constitution appeared. Thus, COFEA provides limited evidence on the meaning of the phrase as it existed prior to its usage in the Constitution.

While identifying broad changes in meaning is relatively easy, inferring the meaning of an individual mention of a word is inherently difficult, and sometimes underdetermined; our judgements as to the tendency of terms in the Constitution to reflect specialized (legal) as opposed to popular meanings should thus be interpreted cautiously. Ultimately, it is inherently difficult to know precisely what an author meant when they chose to use a word in a particular context. Although the combination of frequency differences and differences in inferred meanings suggest the presence of numerous terms in the Constitution that drew heavily on a specialized legal vocabulary, that does not necessarily apply to the Constitution as a whole.

Finally, and most importantly, we do not address broader questions about originalism, including how much it matters what a particular author meant, how a word would have been understood at a particular time, or whether such questions are even relevant to applying the law in a modern context. Independent of legal implications, we hope our work will add to the discussion of limitations associated with corpus linguistics generally, and the use of COFEA for this purpose in particular.

Acknowledgements

Many thanks to Ben Stone, Nicole Coleman, and Dave Armond for facilitating access to COFEA, and to Accessible Archives for providing access to *The Pennsylvania Gazette*. Additional thanks to Candice Laine Penelton for help with initial data exploration, and to Peter Henderson, Dan Jurafsky, and anonymous reviewers for thoughtful feedback on earlier drafts.

References

Stephanie H. Barclay, Brady Earley, and Annika Boone. 2019. Original meaning and the establishment clause: A corpus linguistics analysis. *Ariz. L. Rev.*, 61:505–560.

Dallas Card. 2023. Substitution-based semantic change detection using contextual embeddings. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Mark Davies. 2008. The Corpus of Contemporary American English (COCA). Available online at https://www.english-corpora.org/coca/.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Donald L. Drakeman. 2020. Is corpus linguistics better than flipping a coin? *The Georgetown Law Journal*, 109:81–101.

Matan Eyal, Shoval Sadde, Hillel Taub-Tabib, and Yoav Goldberg. 2022. Large scale substitution-based word sense induction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4738–4752, Dublin, Ireland. Association for Computational Linguistics.

Jonathan Gienapp. 2018. *The Second Creation: Fixing the American Constitution in the Founding Era*. Harvard University Press.

Neal Goldfarb. 2019. A (mostly corpus-based) linguistic reexamination of D.C. v. Heller and the second amendment. *SSRN Electronic Journal*.

Stefan Th. Gries. 2020. Corpora and legal interpretation: Corpus approaches to ordinary meaning in legal interpretation. In Malcolm Coulthard, Alison May, and Rui Sousa-Silva, editors, *The Routledge Handbook of Forensic Linguistics*, 2 edition, chapter 38, pages 628–643. Routledge.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.

Brett Hashimoto. 2023. Corpus of Founding Era American English: designing a corpus for interpreting the United States Constitution. *Corpora*, 18(1):1–14.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Peter Henderson, Daniel E. Ho, Andrea Vallebueno, and Cassandra Handan-Nader. 2024. Corpus enigmas and contradictory linguistics: Tensions between empirical semantic meaning and judicial interpretation. *Minnesota Journal of Law, Science & Technology*, 25(2).

Ethan J. Herenstein. 2017. The faulty frequency hypothesis: Difficulties in operationalizing ordinary meaning through corpus linguistics. *Stan. L. Rev.*, 70.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Matthew Jennejohn, Samuel Nelson, and D. Carolina Núñez. 2021. Hidden bias in empirical textualism. *Georgetown Law Journal*, 109:767–811.

Thomas R. Lee and James C. Phillips. 2019. Data-driven originalism. *University of Pennsylvania Law Review*, 167(2):261–335.

Michael A. Livermore, Felix Herron, and Daniel N. Rockmore. 2024. Language model interpretability and empirical legal studies. *Journal of Institutional and Theoretical Economics (JITE)*, 180(2):244–276.

Gregory E. Maggs. 2012. A concise guide to the records of the Federal Constitutional Convention of 1787 as a source of the original meaning of the Convention of 1787 as a source of the original meaning of the U.S. Constitution. *Geo. Wash. L. Rev*, 1707:1–43.

John O. McGinnis and Michael B. Rappaport. 2018. The constitution and the language of the law. *William and Mary Law Review*, 59(4):1321–1412.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Burt L. Monroe, Michael P. Colaresi, and Kevin M. Quinn. 2017. Fightin' words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403.

Stephen C. Mouritsen. 2010. The dictionary is not a fortress: Definitional fallacies and a corpus-based approach to plain meaning. *Brigham Young University Law Review*, 2010(5):1915–1979.

Clemens Neudecker, Konstantin Baierer, Mike Gerber, Christian Clausner, Apostolos Antonacopoulos, and Stefan Pletschacher. 2021. A survey of OCR evaluation tools and metrics. In *Proceedings of the 6th International Workshop on Historical Document Imaging*

and Processing, HIP '21, page 13–18, New York, NY, USA. Association for Computing Machinery.

Julian Nyarko and Sarath Sanga. 2022. A statistical test for legal interpretation: Theory and applications. *Journal of Law, Economics, and Organization*, 38(2):539–569.

Noah Porter, editor. 1913. *Webster's Revised Unabridged Dictionary*. C. & G. Merriam Co.

Brian G. Slocum and Stefan Th. Gries. 2020. Judging corpus linguistics. *Southern California Law Review Postscript*, 94:13–31.

Lawrence B. Solum. 2019. Originalism versus living constitutionalism: The conceptual structure of the great debate. *Nw. U. L. Rev.*, 113(6):1243–1296.

Uwe Springmann, Dietmar Najock, Hermann Morgenroth, Helmut Schmid, Annette Gotscharek, and Florian Fink. 2014. OCR of historical printings of latin texts: Problems, prospects, progress. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, DATeCH '14, page 71–75, New York, NY, USA. Association for Computing Machinery.

Abigail Stout, Diana Coetzee, and Ute Römer. 2020. "We the citizens?": A corpus linguistic inquiry into the use of "people" and "citizens" in the founding era. *Georgia State University Law Review Georgia State University Law Review*, 36(5):665–697.

Lee J. Strang. 2018. The original meaning of "religion"' in the first amendment: A test case of originalism's utilization of corpus linguistics. *Brigham Young University Law Review*, 2017(6):1683–1750.

Appendix

A Details on Corpora and Sources

A.1 COFEA

COFEA comprises six subcorpora, each of which we briefly describe here. Token counts for each full corpus and the time period of interest are given in Table 3.

EVANS: The Evans Early Imprint Series, created by the Text Creation Partnership, attempts to collect a nearly comprehensive set of books, pamphlets, and broadsides printed in America from the late 17th to early 19th century. As such, it best represents language of more general usage, compared to the other corpora in COFEA.

FOUNDERS: The National Archives Founders Papers Online project collects together the correspondence, personal papers, and private writings from John Adams, Benjamin Franklin, Alexander Hamilton, Thomas Jefferson, James Madison, and George Washington. Although these documents represent much more informal and personal language, compared to the legal documents in COFEA, they are nevertheless unusual for being written by some of the most powerful and privileged people in the United States at that time.

HEIN: COFEA contains a series of collections from HeinOnline, including: the U.S. Treaties and Agreements Library, U.S. Congressional Documents, American Indian Law Collection, and Session Laws Library. As indicated by the titles, these are primarily statutes and legislative records.

ELLIOTS: Compiled and printed by Jonathan Elliot, Elliot's Debates contains the text of debates held at state conventions in relation to adopting the federal Constitution.

STATUTES: Like HEIN, the US Statutes at Large corpus, contains the text of laws enacted by Congress from the end of the 18th century.

FARRANDS: Farrand's Records of the Federal Constitution of 1787 collects together documentary material related to the Constitutional Convention.

A.2 The Pennsylvania Gazette

The Pennsylvania Gazette (TPG) was a prominent U.S. newspaper throughout the 18th century (Strang, 2018). Although it is smaller than the largest COFEA corpora, the digitized version was hand-keyed, and thus avoids many of the problems that arise with OCR, which we leverage in Appendix C.2. We obtained a copy of The Pennsylvania Gazette from Accessible Archives.[8] Token counts for TPG are also given in Table 3.

Subset	Years	Tokens	Tokens (1760-1800)
EVANS	1640–1810	117926408	78327884
FOUNDERS	1706–1836	67070491	41131670
HEIN	1760–1834	43957047	39230854
ELLIOTS	1787	1758037	1758037
STATUTES	1789–1799	531556	531556
FARRANDS	1787	835837	835837
TPG	1725–1815	28671182	19148702

Table 3: Token counts in each subcorpus

A.3 The U.S. Constitution

There are multiple versions of the text of the U.S. Constitution available online; here, we use the electronic copy provided by the National Archives.[9,10] Specifically, we include the preamble and the seven articles, (but not the signatories), along with the Bill of Rights (Amendments I through X). We do not include Article 1 of the 1789 Joint Resolution of Congress (never ratified) or Article 2 (ratified in 1992). Each numbered section of each Article is processed as a separate document (or the entire Article in the case of Articles without numbered sections).

A.4 COCA

The Corpus of Contemporary American English (Davies, 2008) is a large and balanced dataset containing over 1 billion words of American English from multiple genres. Here, we use the text from the academic, fiction, magazine, and newspaper genres, for the years 1990–2017.

B Data Preprocessing

B.1 Initial filtering

Inspecting the documents in COFEA, it is immediately apparent that there are a number of errors, both in terms of metadata, and the text retrieved. For example, several documents in the FOUNDERS corpus consist only of the phrase "> We were unable to find any matches for your search.", which

[8] https://www.accessible.com/accessible/

[9] https://www.archives.gov/founding-docs/constitution-transcript

[10] Interestingly, there are subtle variations in spelling in different versions of the Constitution provided by the websites for different branches of the U.S. government.

is presumably the result of a scraping error. We exclude these documents from our analysis, along with those with the title "Editorial Note", and those with a body less than nine characters long.

To restrict our analysis to those documents that are primarily in English, we use `fasttext` to assess the primary language of each document, and discarded those classified as anything other than English.[11]

B.2 Metadata Assessment and Correction

All documents in COFEA have an associated year. In the case of EVANS, these are typically the year in which the document (such as a broadside) was published. For FARRANDS, by contrast—which provides a record of the Federal Convention of 1787—the documents are associated with the year in which the debates took place, even though this full collection was not published until 1911 (Maggs, 2012). Similarly, the letters in FOUNDERS are typically dated according to the date of the letter.

For the most part, these dates seem adequate to historical investigation. One collection, however, deserves additional scrutiny, namely HEIN. Although HEIN is one of the three large collections in COFEA by number of tokens, it consists of only 285 documents, in the version of COFEA we gained access to. Obviously many of these documents are quite long, including entire books. In this case, using the date of publication presents more of an issue, if the goal is to capture the language of a particular era. In particular, some documents in HEIN are historical texts, but include a modern preamble. In other cases, the documents collect together laws that were written over many years, such as "Laws of the State of New York".

We briefly reviewed all of the documents in HEIN, and try to associate each document with the most recent text within it, to avoid contaminating historical text with modern usages. Fortunately, the given dates seem to be mostly reliable with respect to whether documents were written during the time period 1760–1800 or not. Thus, the issue with dates has very little effect on our selection of documents to study. However, a somewhat larger number of documents appear to have potentially unreliable dates relative to the year in which the Constitution was written (i.e., they may have been written before 1787, but dated later, or vice versa). As such, one should be especially cautious when trying to look for usages which pre-date the writing of the Constitution itself.[12]

B.3 Spelling normalization

Because some terms appear with varied or archaic spellings (e.g., "choose" vs "chuse"), we begin by normalizing the spelling of all occurrences of terms that appear in the Constitution for which we can find a relevant alternate form. In more detail, we first use word2vec (Mikolov et al., 2013) to train static word vectors on documents from COFEA (from all years), after converting all text to lower case. For all terms in the Constitution, we find the top 100 most similar word vectors, in terms of cosine similarity, and compute the edit distance between the corresponding pair of words. We filter out those that have cosine similarity less than 0.5 or an edit distance of more than two. We then manually inspect all candidates for each term, and keep those that can be identified as legitimate alternate spellings. We then replace all version of the alternate spellings everywhere in our corpora with a standardized form. Ultimately, at most one alternate spelling was found for each term, mostly reflecting differences between British and American spellings. The full list of alternate spellings identified is given in Table 4.

B.4 Identifying bigrams

Because past work has focused on certain bigrams in the Constitution (Lee and Phillips, 2019), we curate a small list of multi-word expressions, to which we add past bigrams that have been considered. To construct our list, we compute the normalized pointwise mutual information (NPMI) for all bigrams that appear in COFEA (not just in the Constitution). We then keep noun-noun and adjective-noun bigrams with NPMI ≥ 0.6 and at least 2000 mentions in COFEA. We also exclude bigrams with incorrectly tagged parts of speech, as well as names of people, quantities, streets, cities, counties, or bodies of water, and greetings, such as "obedient servant". Finally, we add in the three bigrams from Lee and Phillips (2019) for comparison with past work, even though they would not meet our selection criteria for inclusion. The resulting bigrams are given in Table 5, along with the ones added from past work.

[11] https://fasttext.cc/docs/en/language-identification.html

[12] Note that the token counts per year showing in Figure 1 uses the uncorrected dates.

Word	Alternate spelling
ambassadors	embassadors
among	amongst
authorized	authorised
behavior	behaviour
cannot	canot
choose	chuse
choosing	chusing
compel	compell
control	controul
controversy	controversie
days	dayes
defense	defence
domestic	domestick
increase	encrease
entered	entred
expel	expell
favor	favour
guarantee	guaranty
habeas	habeus
honor	honour
inferior	inferiour
judgment	judgement
labor	labour
limited	limitted
massachusetts	massachusets
misdemeanors	misdemeanours
needful	needfull
net	nett
offense	offence
offenses	offences
organizing	organising
payment	paiment
pennsylvania	pensylvania
piracies	pyracies
privilege	priviledge
privileged	priviledged
privileges	priviledges
public	publick
receive	recieve
repel	repell
rhode	rhoad
secrecy	secresy
soldier	souldier
supreme	supream
swear	sware
tranquility	tranquillity
trial	tryal
tried	tryed
until	untill
useful	usefull
vessels	vessells
welfare	wellfare
writs	writts

Table 4: Alternate spellings identified for normalization

Bigram	Tags	NPMI	Count
nova scotia	NN	0.99	3010
united states	NN	0.93	145086
rhode island	NN	0.88	14042
new york	NN	0.82	66155
south carolina	NN	0.82	15966
west indies	NN	0.82	8809
head quarters	NN	0.78	12090
reasonable charges	JN	0.78	8750
u. s.	NN	0.76	3466
north carolina	NN	0.76	12787
yellow fever	JN	0.75	2291
small pox	JN	0.75	4602
minister plenipotentiary	NN	0.75	2930
fellow citizens	JN	0.73	8278
great britain	NN	0.72	35462
common pleas	NN	0.71	4388
west india	NN	0.70	3965
new jersey	NN	0.70	16867
new hampshire	NN	0.68	10289
continental congress	NN	0.67	12059
east india	NN	0.67	2446
lieutenant colonel	NN	0.66	2819
court martial	NN	0.66	5150
grand jury	NN	0.65	2101
military stores	JN	0.64	2656
indian corn	NN	0.64	2149
dwelling house	NN	0.63	6707
human race	JN	0.63	2006
fellow creatures	NN	0.62	2088
pounds sterling	NN	0.62	2236
quarter master	NN	0.61	4532
holy scriptures	NN	0.61	2669
foreign affairs	NN	0.60	4620
french republic	NN	0.60	3219
vice president	NN	0.60	3998
domestic violence	JN	0.36	92
natural born	JN	0.35	368
public use	JN	0.20	738

Table 5: Bigrams included in our analysis

C OCR Quality Assessment

C.1 Dictionary-based OCR assessment

To assess the quality of documents according to coverage in an appropriate dictionary, we make use of the 1913 edition of Webster's Revised Unabridged Dictionary (Porter, 1913), which is available in tabular form online.[13] Because dictionaries typically do not include proper names, we augment the list of terms in this dictionary with the names of all countries, as well as common names of U.S. cities, states, and counties.[14] We additionally add in titles (e.g., "mr", "mrs", "esq") and common person names, including names from the Old Testament,[15] and common baby names based on data from the U.S. Social Security Administration.[16]

We then preprocess each document in COFEA and TPG by using spaCy (Honnibal and Montani,

[13] https://github.com/ahacop/websters-dict-1913-stardict

[14] https://github.com/grammakov/USA-cities-and-states/

[15] https://github.com/hadley/data-baby-names/blob/master/old-testament.txt

[16] https://github.com/hadley/data-baby-names/blob/master/baby-names.csv

2017) to convert all words to lemmas. From this, we drop punctuation, and convert tokens to lowercase. Based on inspecting the remaining tokens, we additionally add to the dictionary common abbreviations for months (e.g., "feb"), roman numerals (e.g., "xvi"), ordinals (e.g., "1st"), common abbreviations (e.g, "servt"), and obvious misspellings (e.g., "recieve"). Given these exclusions and adjustments, we finally compute the proportion of remaining words in each document that appear in the dictionary. In doing so, we further allow for certain suffixes that may have been missed by the lemmatizer (e.g., "ly"), as well as accommodating common British spellings (e.g., substituting "our" for "or" in a word). For full details, please refer to online replication code.

C.2 Language model OCR assessment

As an alternative way of assessing OCR quality, we also try using a character language model to estimate the perplexity of each document. Using `kenlm` (Heafield, 2011), we train a trigram character language model on the union of TPG and the similarly hand-keyed (Text Creation Partnership) portion of the Eighteenth Century Collections Online project (ECCO-TCP), both of which can be assumed to be of high quality.[17] We then use this model to assess the perplexity of each document in each subcorpus in COFEA. Because `kenlm` is designed for word-level models, we convert all documents to sequences of characters, separated by spaces, with spaces converted to a `<space>` token. We also use the "discount fallback" option, because the distribution of characters differs dramatically from that of words.

The results of this assessment are show in Figure 4. As with the dictionary-based assessment, we find that the HEIN subcorpus again appears to be among the worst, in so far as the best documents (farthest to the right) in HEIN are worse than the best document in other collections. Surprisingly, FOUNDERS and FARRANDS appear to be worse than HEIN in the worst cases. In the case of FARRANDS, this appears to be due to the fact that some documents are very short, and contain a high number of capitals letters and punctuation (which are improbable given our training documents), such as "SATURDAY, JUNE 9, 1787". More generally, the presence of abbreviations, symbols, and roman numerals referring to other sections also hurts perplexity.

[17] https://textcreationpartnership.org/tcp-texts/

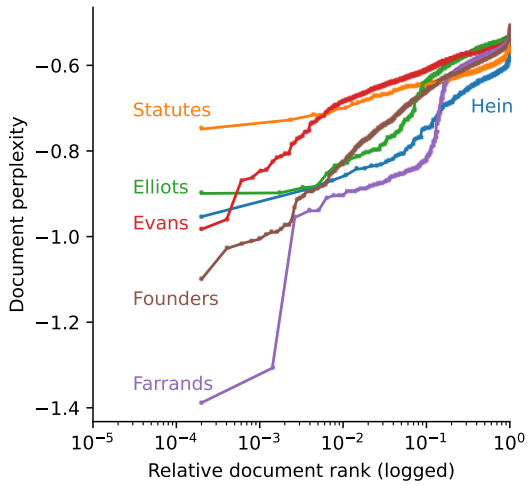


Figure 4: OCR quality assessment made using a trigram character language model.

The low quality documents in FOUNDERS can similarly be attributed to very short and terse documents (e.g., "July 31. 13. 25.lb brown sugar Th:J."), as well as text that does not resemble normal paragraphs of text (e.g., lists of names or citations). The key takeaway here is that these documents do not seem to have issues with the transcription (such that keyword searches should not be affected), but that certain terms may occur in unusual contexts in these documents.

C.3 Word frequencies across subcorpora

As a final way of identifying potential OCR problems, we examine raw frequency differences between collections in COFEA. Using the log-odd technique from Monroe et al. (2017), we identify the tokens that are most over-represented in each of the three largest collections (EVANS, FOUNDERS, and HEIN), relative to the others.

From this analysis, it is clear that there are differences between the COFEA collections in terms of formality, orthography, languages used, and OCR quality. For example, the most frequent unique terms in EVANS are distinctive punctuation (e.g., •, !, >, <, ?, "), religious terms (e.g., *God*, *Christ*, *church*), and archaic forms (e.g., *thy*, *thou*, *ye*). Distinctive punctuation also appears in FOUNDERS (e.g., & -, ", ", [,]), as well as terms of address (e.g., *sir*, *excellency*), and various short forms (e.g., *obedt*, *servt*, *genl*, *yr*).

For HEIN, many of the most distinctive terms are a mix of governmental terms (e.g, *court*, *commit-*

Term	Count
shall	137392
thall	29932
fliall	12263
ihall	11966
flall	7633
flhall	5527
lhall	3111

Table 6: Frequent misspellings of the term "shall" in HEIN, illustrating the prevalence of OCR errors.

tee, *report*), and apparent OCR mistakes, revealing differences in how these different datasets were digitized. For example, multiple of the most frequent distinctive terms appear to be misspellings of the word "shall", such as "fhall" (see Table 6). This likely results in part from the use of the "long S", which many OCR engines may misinterpret as an "f" or "t". In other words, we should be aware that the documents from each subset may be partially identifiable based purely on orthographic features, independent of content.

D Measuring Change in Meaning

As described in the main paper, we measure meaning change by getting probable replacements for masked terms, and looking at how the distribution of these replacements differs between periods (or collections). As per Eyal et al. (2022), we first adapt a pretrained language model to the data, by doing continued masked language model pretraining, starting with `bert-large-uncased`. In particular, we continue training for five epochs, either on the union of COFEA and COCA, or just COFEA, for measuring change and variation, respectively. We then index the relevant corpora, to collect all occurrences of each term, and sample up to 4000 occurrences of each. We include all the terms in the Constitution (treating bigrams in Table 5 as single words), along with 10,000 random background terms, selected using stratified sampling, to over-represent more common tokens. In addition, for background terms, we limit ourselves to those with at least 50 occurrences, minimum length of 2, and exclude punctuation.

To get replacements, we mask the term of interest (replacing with it with a single mask token, even if it is composed of multiple tokens), and provide the mask, along with up to 50 tokens of context to either side of the fine-tuned model. We then collect the top 10 most likely replacement tokens, only counting whole word replacements with length greater than one, and excluding stopwords from the Snowball sampler. To measure meaning change between two corpora, we collect all replacements for each word from each corpora (e.g., COFEA vs. COCA), and measure the Jensen-Shannon divergence of the two distributions. All processing was run on one A6000 GPU.

Because there is no definitively best method for assessing changing in meaning, we also run an evaluation using the word vector-based approach from Hamilton et al. (2016). In particular, we use `gensim` to fit 100-dimensional word2vec vectors, with a window size of 5, for each corpus or collection to be compared. We then align word vectors using the Procrustes alignment, and use the cosine distance between vectors for a given word as an estimate of the difference in meaning.

D.1 Change in Meaning Over Time

As described in the main paper, we first measure change over time, by comparing COFEA to COCA using the method described above. A more complete list of the terms with the greatest change in meaning, is given in Table 7, along with corresponding JSD values.

To assess the amount of change in meaning experienced by terms in the Constitution, we compare these to the sampled set of random background terms mentioned above. Because of the inverse correlation between meaning change and term frequency noted in past work (Hamilton et al., 2016; Card, 2023), we attempt to measure the association with being in the Constitution while accounting for frequency. In particular, we model change in meaning using linear regression as a function of logged term frequency, and whether the term is in the Constitution. More precisely, we fit,

$$\text{JSD}_t = \beta_0 + \beta_1 \cdot \log(\text{count}_t) + \beta_2 \cdot \mathbb{I}[t \in C] + \epsilon_t,$$

using all terms in the Constitution, along with a large set of background terms. JSD_t is the JSD for term t, count_t is the count of term t in both COFEA and COCA combined, ϵ_t is the error associated with term t, and C is the set of terms in the Constitution. The output of this regression is given in Table 8, showing that the coefficient associated with constitutional terms is indeed significant, although the effect is small.

Figure 5 shows the measured change for constitutional terms (in orange) compared against this random sample of other terms (in blue), with select

Term	Founding era	Modern era	JSD
captures	captures prizes capture seizures prize	captures reflects shows represents describes	0.91
domestic violence	invasion insurrection violence invasions	violence abuse rape crime assault	0.91
marque	mar truce protection war commission	porte salle junta grange crescent	0.90
capitation	direct poll land general state	medicare payment insurance compensation	0.90
quartered	stationed posted kept placed lodged	sliced chopped peeled seeded trimmed	0.86
affirmation	declaration oath certificate deposition	expression recognition acceptance assertion	0.86
training	training raising bringing exercising trained	training education instruction practice	0.85
piracies	crimes offenses piracy murders treason	crimes crime abuses offenses acts	0.85
natural born	natural native free good born	serial cop born psycho professional	0.85
arsenals	magazines fortifications stores barracks	weapons forces capabilities arsenal arms	0.83
emolument	advantage profit benefit interest happiness	finance money profit advantage brown	0.83
counterfeiting	altering making printing destroying signing	fraud theft crime smuggling terrorism	0.83
presentment	complaint trial information indictment report	consolidation verdict processing hearing	0.82
test	test trial proof late bankrupt	test tests testing exam assessment	0.82
reprisal	commissions commission protection passports	retaliation retribution violence punishment	0.82

Table 7: Constitutional terms with the largest meaning change from the founding to the modern era, shown with most common substitutes, and corresponding JSD values.

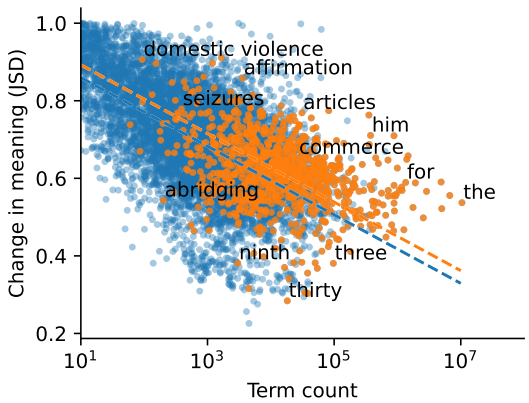


Figure 5: Change in meaning between founding and modern eras vs. term counts in both corpora combined, with constitutional terms shown in orange, random background terms in blue, and select terms labeled.

Variable	Coefficient	Std. err.	p-value
Intercept	0.9480	0.003	< 0.001
In Constitution	0.0328	0.004	< 0.001
log(Count)	-0.0384	< 0.001	< 0.001

Table 8: Regression results modeling change in meaning (JSD) as a function of term frequency and whether or not the term is in the Constitution. As shown, constitutional terms have changed slightly but significantly more than random background terms since the Founding era.

terms labeled. It also shows the fitted regression model, giving an intuitive sense of the difference between constitutional and background terms with respect to change in meaning.

For comparison, Table 9 shows the 15 terms with the greatest change over time, according to the method of Hamilton et al. (2016). This list has many broad similarities with the results of our method, but also some differences. Four terms

Term	Cosine Distance
captures	1.19
respecting	0.90
whereof	0.87
likewise	0.84
vest	0.84
compact	0.83
natural born	0.82
writ	0.82
erection	0.79
emolument	0.79
magazines	0.79
capitation	0.77
compulsory	0.77
corpus	0.77
cannot	0.76

Table 9: Constitutional terms with the largest meaning change from the founding to the modern era, according to the method of Hamilton et al. (2016).

overlap with the top 15 from Table 7. Other terms here are similarly ranked by both methods (e.g., *magazines* is ranked 20th by the method in our paper). Surprising here is the inclusion of *likewise* (ranked in the bottom half of terms by the method in our paper), and the absence of *domestic violence* (which has clearly changed dramatically over time). Overall, the results from the two metrics have a correlation of 0.64.

D.2 Specialized vs. Popular Meanings

As described in the main paper, we assess differences in meaning between legal vs. popular sources by using the method described above to compare legal sources in COFEA (FARRANDS, ELLIOTS, STATUTES, and the documents in HEIN marked as "Legal") vs. popular sources (EVANS, and The Pennsylvania Gazette; TPG). The top 15 terms with the largest difference in meaning across sources are

Term	Legal	Popular	JSD	Lean
aid*	aid said laid id hid	aid assistance support help protection	0.80	P
tender	tender payment demand currency money	tender kind soft generous great	0.80	L
fix*	fix three four two five	fix establish set determine settle	0.78	P
dock	dock ship navy naval docks	dock market water street front	0.75	L
fled*	fled ed led ted fed	fled retired escaped returned went	0.72	I
bankruptcies	commerce trade religion slaves debts	losses debts commerce trade failures	0.71	L
affecting	affecting respecting touching concerning	affecting interesting awful melancholy	0.70	L
repassed	rejected amended approved repealed	passed crossed entered left ascended	0.70	L
resignation	resignation removal refusal appointment	resignation submission patience obedience	0.70	L
sign	sign receive make deliver take	sign head tavern foot signs	0.69	L
searches	search searches seizures arrests attacks	searches search knows sees inquiries	0.69	L
domestic	domestic national public foreign internal	domestic private public social family	0.69	I
escaping	escaping returning flying taken escape	escaping escape avoiding returning escaped	0.68	I
vest	vest invest vested place leave	vest coat jacket shirt hat	0.67	L
training	training muster organizing regulating exercising	training raising bringing trained building	0.67	L

Table 10: Constitutional terms with the largest difference in meaning between legal and popular sources in COFEA, shown with most common substitutes, and corresponding JSD values. Terms marked with an asterisk (*) appear to be primarily due to common OCR errors in Hein. The Lean column shows whether the specific mentions of the term in the Constitution appear to lean more towards the typical legal meaning (L), popular meaning (P), or are indeterminate (I).

shown in Table 10, with those which seem to be due to OCR error in HEIN marked with an asterisk. As can be seen, while some differences are related to different syntactic forms (e.g., *sign* as a verb as opposed to a noun), others show a subtle variation in meaning between legal and popular sources (e.g., *resignation*, *searches*, *domestic*, *escaping*).

For comparison, Table 11 shows the 15 terms with the greatest meaning variation across sources, according to the method of Hamilton et al. (2016). These results, compared against Table 10 are again broadly similar. This time, there are six terms in common among the top 15, but the correlation between scores is only 0.53. As with the results in the main paper, the lower values here again indicate less variation across sources within COFEA, relative to change over time.

Term	Cosine Distance
affecting	0.87
tender	0.81
bankruptcies	0.79
sign	0.79
discoveries	0.76
training	0.76
confession	0.71
sundays	0.70
aid	0.70
high	0.70
likewise	0.69
privileged	0.68
reexamined	0.67
comfort	0.67
nobility	0.66

Table 11: Constitutional terms with the largest meaning change from the founding to the modern era, according to the method of Hamilton et al. (2016).

D.3 Assessing meaning in the Constitution

Although inferring the meanings of individual mentions of terms is inherently difficult, we attempt to characterize the broad meanings of terms as they occur in the Constitution. To do so, we rely on the overall most common meanings inferred from comparing the legal to the popular sources.

For each mention of a term in the Constitution, we gather the k most common replacements, (with $k = 10$), and then compare this set to the corresponding set of k overall most common replacements for each subset (legal vs. popular). That is, let R_c^w be the set of ten most common replacements for word w from corpus c, for $c \in \{\text{Legal } (L), \text{Popular } (P)\}$. For each mention of w in the Constitution, we use the same process to collect the ten most probable replacements for that specific mention, which we denote R_m^w. We then compute the overlap between the set for that mention with corresponding replacements set for legal and popular sources. That is, let $o_c^w = |R_m^w \cap R_c^w|$. If $o_L^w > o_P^w + 1$, then we count that mention of word w as a specialized legal usage. If $o_P^w > o_L^w + 1$, then we count it as a popular usage. If neither is satisfied, we count it as indeterminate.

This gives us an imperfect but still useful measure of meaning alignment. For example, consider the word *tender*. This term has numerous meanings, encompassing nouns, verbs, and adjectives. In the popular sources, the dominant meaning has to do with kindness (kind, soft, generous, great). In

legal sources, by contrast, the dominant meaning has to do with payment (payment, demand, currency, money, security). The word occurs once in the Constitution ("No State shall . . . make any Thing but gold and silver Coin a Tender in Payment of Debts"), from which we can infer that it is being used in the sense of payments. This too is captured by our metric; the top replacements suggested by the model (medium, fund, currency, standard) do not perfectly align with either usage, but are ultimately more similar to the financial sense, which is more common in legal documents, and thus $o_L^{\text{tender}} > o_P^{\text{tender}} + 1$. The same applies to terms like *faith* and *sign*, although many are less clear cut.

D.4 Effect of Continued MLM Training

Unsurprisingly, because BERT was trained primarily on contemporary text, it is better at predicting masked words from modern documents (i.e., COCA), rather than founding era documents. To mitigate this issue, we make use of continued masked language model training, to adapt the base model to our data, as described in the main paper. As a way of quantifying the impact of this, we compute the proportion of mentions included in our analyses in which the original (masked) word is included in the set of predicted terms, and then compare this to using the corresponding off-the-shelf model.

Using the vanilla `bert-large-uncased` model, (without continued training), we find that there is indeed an imbalance between the early and modern text. The proportion of mentions in which the masked term is included in the set of 10 most likely predictions is 60.0% in the modern documents and only 40.0% in the founding era documents. When using the model that has been adapted to these data (which is what we use for obtaining the results reported in this paper), the corresponding values are 64.6% and 60.0%, respectively, demonstrating that the continued MLM training has been effective in adapting the model to our data.

MindRef: Mimicking Human Memory for Hierarchical Reference Retrieval with Fine-Grained Location Awareness

Ye Wang[1*], Xinrun Xu[2,3*], Zhiming Ding[3†]
[1]Renmin University of China
[2]University of Chinese Academy of Sciences
[3]Institute of Software, Chinese Academy of Sciences
yewang@ruc.edu.cn, xuxinrun20@mails.ucas.ac.cn, zhiming@iscas.ac.cn

Abstract

When completing knowledge-intensive tasks, humans sometimes need an answer and a corresponding reference passage for auxiliary reading. Previous methods required obtaining pre-segmented article chunks through additional retrieval models. This paper explores leveraging the parameterized knowledge stored during the pre-training phase of large language models (LLMs) to recall reference passage from any starting position independently. We propose a two-stage framework that simulates the scenario of humans recalling easily forgotten references. Initially, the LLM is prompted to recall document title identifiers to obtain a coarse-grained document set. Then, based on the acquired coarse-grained document set, it recalls fine-grained passage. In the two-stage recall process, we use constrained decoding to ensure that content outside of the stored documents is not generated. To increase speed, we only recall a short prefix in the second stage, and then locate its position to retrieve a complete passage. Experiments on KILT knowledge-sensitive tasks have verified that LLMs can independently recall reference passage locations in various task forms, and the obtained reference significantly assists downstream tasks. [1]

1 Introduction

Knowledge-intensive tasks rely heavily on large knowledge sources (Petroni et al., 2021). Traditional methods often use retrieval models to find relevant passages from resources like Wikipedia for tasks such as question answering (Izacard and Grave, 2021). However, limitations exist with both sparse (lack of semantic depth) and dense retrieval (limited interaction between question and passage representations) (Khattab et al., 2021). Generative retrieval methods, leveraging models' generative abilities for deeper interaction with knowledge sources, are gaining popularity (Tay et al., 2022). However, Current retrieval methods require pre-segmented passages, limiting reference flexibility like human memory. We ask: "Can LLMs bypass chunking to recall references from any position?"

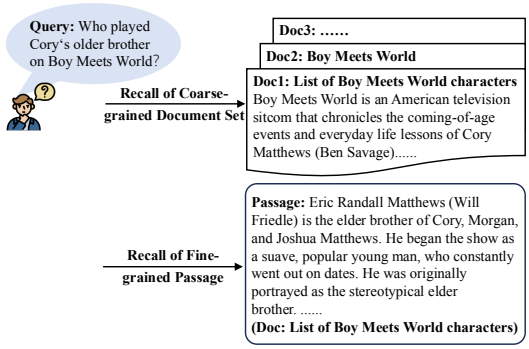


Figure 1: Human recall of forgotten information often involves a two-step process: recalling memorable documents first, then locating the specific passage within.

Leveraging the capabilities of LLMs, we propose **MindRef**, a two-stage framework for flexible passage retrieval. Inspired by human recall, we first prompt the LLM to recall relevant document titles, guided by a Trie (Cormen et al., 2022). Then, using an FM-index (Ferragina and Manzini, 2000) built from the retrieved documents, the LLM recalls specific passages with flexible starting points. A weighted score combines both stages for final reference selection. To enhance efficiency, MindRef retrieves passages by recalling only a short prefix. The LLM generates this prefix, which is located within the documents using the FM-index and KMP algorithm. Ultimately, the algorithm identifies a longer passage as the final reference. This approach allows LLMs to access and retrieve natural references from articles of any length without relying on additional retrieval models or pre-segmentation, offering both flexibility and effi-

[*]Equal Contribution.
[†]Corresponding Author.
[1]Code is available at https://github.com/www-Ye/MindRef.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 857–872
July 27 - August 1, 2025

ciency.

Extensive experiments on 6 KILT benchmark tasks (Petroni et al., 2021) demonstrate the effectiveness of **MindRef**, enabling open-source LLMs like LLaMA (Touvron et al., 2023a) and LLaMA-2 (Touvron et al., 2023b) to retrieve documents and passages, improving downstream task performance effectively. Key contributions include:

1) We propose MindRef, a cognitively-aligned retrieval framework that formalizes the 'document-to-detail' mechanism of human memory into neural architectures for the first time. 2) Breaking away from pre-chunking paradigms, our method achieves chunkless reference localization through joint Trie-FMIndex constrained decoding, enabling retrieval from arbitrary document positions. 3) The SPRL co-optimization strategy delivers 4× faster inference while maintaining 95%+ accuracy via short-prefix neural recall. 4) Extensive evaluations on 6 KILT tasks demonstrate state-of-the-art performance, with MindRef-boosted LLaMA-2 achieving 78.79% accuracy on FEVER.

2 MindRef Framework

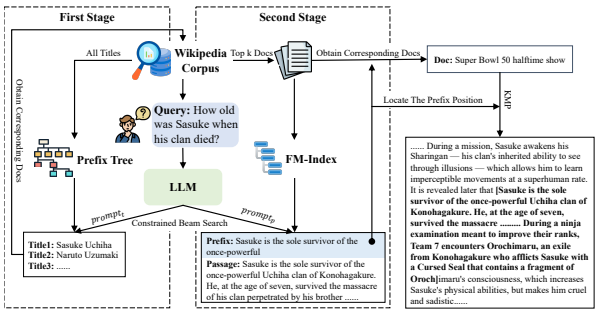


Figure 2: **MindRef** Framework. First, all Wikipedia titles are stored in a prefix tree, then the LLM is prompted to recall title identifiers under this prefix tree constraint. Subsequently, an FM-index is constructed from the top k documents obtained, and the LLM recalls reference passage under the new constraint.

In this section, we detail our two-stage framework, **MindRef** (Figure 2). In the first stage, we prompt the LLM to recall title identifiers, which serve as candidate documents for the next stage. In the second stage, the LLM is prompted to recall reference passage from the documents obtained in the first stage. To increase speed, we only recall a short prefix, then locate and extract the reference within the document. Detailed Prompt can be found in Subsection B.2.

2.1 Stage 1: Coarse-Grained Document Recall

Recalling fine-grained reference passages directly for knowledge-intensive tasks can be challenging (Subsection B.3). Therefore, we propose a two-stage process. First, we retrieve easily-recallable documents (e.g., Wikipedia pages) by leveraging their titles as unique identifiers. Using a Trie (Section A.1) data structure (Cormen et al., 2022), we prompt the LLM to recall relevant titles, ensuring they correspond to existing pages.

Given a query x and prompt $prompt_t(x)$ (e.g., "Question: \n \n The Wikipedia title corresponding to the above question is: \n \n Title:"), the LLM generates titles, guided by the Trie. The Trie, based on previously generated tokens, restricts subsequent token choices to valid prefixes within the set of all Wikipedia titles (T), effectively guiding the LLM along valid title paths.

This first stage focuses on efficiently retrieving a set of candidate documents before proceeding to fine-grained passage retrieval within these documents in the second stage. The score for generating title t given prompt $prompt_t(x)$ is calculated using the standard implementation from the library:

$$\begin{aligned} & score_1(t|prompt_t(x)) \\ &= \frac{\log p_\theta(y_t|prompt_t(x))}{|y_t|} \\ &= \frac{\sum_{i=1}^{l_t} \log p_\theta(y_i|y_{<i}, prompt_t(x))}{l_t} \end{aligned} \tag{1}$$

where y_t represents the set of tokens in title t, l_t and $|y_t|$ represent the number of tokens used to generate the title, θ represents the model's parameters.

2.2 Stage 2: Fine-Grained Passage Recall

Following the initial identification of relevant Wikipedia pages, we move to fine-grained passage retrieval within those documents. We employ the FM-index ((Section A.2)) constraint (Ferragina and Manzini, 2000), a space-efficient data structure enabling fast substring search and supporting retrieval from arbitrary positions.

After obtaining the top k titles and their documents (D_k), we construct a targeted FM-index specifically for D_k, reducing the search space. These FM-indexes are pre-built for all documents to avoid on-the-fly construction.

We then prompt the LLM with $prompt_p$ (e.g., "Question: \n \n The answer to the above question can be found in the following Wikipedia paragraph:\n \n Answer:") to generate a passage p. The FM-index, based on previously generated tokens,

dynamically provides permissible successor tokens, guiding the LLM to generate valid passages from any position within D_k. We measure the score of the task corresponding passage by using the autoregressive formula to calculate the score:

$$\begin{aligned} & score_2(p|prompt_p(x)) \\ &= \frac{\log p_\theta(y_p|prompt_p(x))}{|y_p|} \\ &= \frac{\sum_{i=1}^{l_p} \log p_\theta(y_i|y_{<i}, prompt_p(x))}{l_p} \end{aligned} \tag{2}$$

where y_p represents the set of tokens in the passage p, θ is the model parameters, $|y_p|$ and l_p represent the number of tokens generating the passage, usually set between 150 to 200. To integrate information generated from both stages, we calculate the weighted sum of the scores from the first and second stages to obtain the final score under input query x:

$$\begin{aligned} score(p|x) &= \alpha * score_1(t|prompt_t(x)) \\ &+ (1-\alpha) * score_2(p|prompt_p(x)) \end{aligned} \tag{3}$$

where $score_1(t|prompt_t(x))$ is the score of the Wikipedia page title t corresponding to the passage p, α is a hyperparameter controlling the weight of the two stages. Finally, among all recalled passages, the one with the highest $score(p|x)$ value is selected as the best reference.

2.3 Short Prefix Recall and Localization

While LLMs excel at recalling long passages, their inference speed hinders practical application. To address this, we propose Short Prefix Recall Location (SPRL), aiming to locate passages by recalling only a short prefix.

Initially, given a question q, SPRL prompts the LLM to generate a short prefix p_s of length l_{p_s} using the same prompt ($prompt_p$) as in the second stage, significantly reducing generation cost. Subsequently, SPRL attempts to identify the document d containing p_s within the document set D_k obtained in the first stage. Due to the limited size of D_k, p_s typically maps to a unique document d. If p_s is found in multiple documents within D_k, the first such document encountered is selected by default, ensuring a deterministic outcome. Next, using the KMP algorithm, the first starting position st of p_s in d is determined, and a complete reference passage $p_{final} = d[st : st + l_p]$ is extracted. The final score is calculated using Equations 2 and 3 to select the best reference. Experiments (Section 3.2.1) demonstrate that recalling only the prefix for localization yields effective results.

3 Experiments

In this section, we conduct comprehensive experiments on coarse-grained pages, fine-grained passage-level reference evaluation, and downstream tasks to validate the effectiveness of our framework. Additionally, we perform further analyses and experiments in the Appendix through Further Analysis, and Case Studies.

3.1 Experimental Setup

Datasets: Experiments were conducted on 6 knowledge-sensitive tasks from the KILT benchmark (Petroni et al., 2021) ((Section B.1)).

Evaluation Metrics: R-Precision for page-level retrieval. Answer in Context (percentage of references containing at least one gold answer) for NQ, TriviaQA, and HotpotQA. Entity in Context (percentage of references containing at least one gold entity) for other datasets. Downstream task metrics followed the official KILT implementations.

Baseline Models: We compare with several traditional retrieval models. These models all use the passage segmentation from the official KILT as the source for obtaining reference. For unsupervised retrieval models, we compare the traditional sparse retrieval model BM25[2] (Robertson and Zaragoza, 2009), and the dense retrieval model Contriever (Izacard et al., 2022). We also compare with the dense retrieval model DPR[3] (Karpukhin et al., 2020) that has been fine-tuned on the full dataset. We input the first passage retrieved by the model as the reference context into the LLM, which then reads the relevant reference to answer downstream tasks.

Implementation Details: LLaMA (Touvron et al., 2023a) and LLaMA-2 (Touvron et al., 2023b) (7b and 13b) were used for reference recall. LLaMA-2-13b served as the reading model for downstream tasks. We merge the passage fragments from KILT into complete documents, serving as the data source for recall. The length of the complete documents is arbitrary. In the recall phase, we always use a beam search generation strategy. In the first stage of generation, the beam size is set to 15, and we construct an FM-index containing the top $k = 2$ documents. In the second stage, the beam size is set to 10, the length of

[2]We implement BM25 retrieval using the https://github.com/castorini/pyserini repository

[3]We conduct experiments with the trained DPR model and preprocessed vector index from the https://github.com/facebookresearch/KILT repository.

Table 1: Coarse-grained page-level results (R-Precision). ⋆ denotes full data training. Cyan indicates best results, and pink indicates second-best.

Method	Open-domain QA				Fact Check.	Dial.
	NQ	TriviaQA	HotpotQA	ELI5	FEVER	WoW
Contriever	34.72	34.28	26.14	11.02	55.64	29.67
BM25	26.33	31.78	41.30	6.83	52.09	28.78
DPR⋆	54.74	45.68	25.46	**16.19**	56.61	26.62
MindRef (LLaMA-7b)	54.46	**57.03**	44.56	**15.13**	76.57	**52.91**
MindRef (LLaMA-13b)	54.42	55.53	**46.30**	12.94	**77.55**	34.51
MindRef (LLaMA-2-7b)	**56.33**	**56.43**	46.20	14.60	77.29	49.61
MindRef (LLaMA-2-13b)	**57.77**	54.41	**48.70**	15.00	**83.69**	**57.63**

Table 2: Fine-grained passage-level results (Answer/Entity in Context for top-1 reference).

Method	Open-domain QA				Fact Check.	Dial.
	NQ	TriviaQA	HotpotQA	ELI5	FEVER	WoW
	Answer in Context			Entity in Context		
Contriever	19.28	37.21	11.16	12.48	40.48	45.15
BM25	23.65	58.87	**29.45**	12.01	**58.33**	50.36
DPR⋆	**47.94**	**66.60**	20.29	14.40	41.22	45.38
MindRef (LLaMA-7b)	36.87	58.48	25.55	**15.99**	54.85	**59.40**
MindRef (LLaMA-13b)	37.72	60.96	26.34	14.80	55.20	50.79
MindRef (LLaMA-2-7b)	38.07	62.88	27.55	**16.85**	56.23	57.79
MindRef (LLaMA-2-13b)	**40.82**	**68.20**	**30.04**	15.06	**58.42**	**63.43**

the short prefix is $l_{p_s} = 16$, and we extract a token length of $l_p = 150$ as the final reference. The weight setting for the two-stage weighted method is $\alpha = 0.9$. All downstream tasks use greedy decoding. The prompts used in the experiments can be found in Appendix B.2. Experiments were conducted on Tesla A100 40G GPUs.

3.2 Experimental Results

3.2.1 Page-level Results

Coarse-grained page-level results, as shown in Table 1, demonstrate that the MindRef framework, when implemented with Llama-2-13b, achieves the best R-precision scores of 57.77, 48.70, 83.69, and 57.63 on the NQ, HotpotQA, FEVER, and WoW datasets, respectively. This significantly surpasses the performance of sparse retrieval BM25 and dense retrieval Contriever in a zero-shot scenario. It also shows strong competitive power against the fully trained DPR method, especially on the WoW and FEVER datasets, with improvements of 27.08 and 31.01 points, respectively. This result is consistent with the hypothesis that LLMs are powerful in recalling coarse-grained title identifiers, enabling the acquisition of high-quality relevant pages that assist in the subsequent fine-grained recall stage.

3.2.2 Passage-level Results

Fine-grained reference passage results, as shown in Table 2, reveal that the MindRef framework, when implemented with Llama-2-13b, also achieves the best scores of 68.20, 30.04, 58.42, and 63.43 on the TriviaQA, HotpotQA, FEVER, and WoW datasets, respectively. We note that the improvement of the framework in fine-grained reference passage compared to the DPR method is relatively reduced compared to the page-level results. This suggests potential for optimization in activating LLMs to recall more detailed and longer reference, presenting a greater challenge compared to recalling shorter title. Notably, DPR performs excellently on the NQ dataset, which may relate to its training data format. Interestingly, in the HotpotQA dataset, BM25 remains competitive, surpassing dense retrieval methods, possibly due to the longer questions in this dataset leading to more vocabulary overlap. MindRef shows significant progress on the FEVER and WoW datasets, demonstrating the potential and adaptability of LLMs in recalling high-quality reference passage across different task formats. Furthermore, the general enhancement in performance with the progression from Llama to Llama-2 and the increase in model size indicates a correlation between the recall ability and the underlying capabilities of LLMs.

Table 3: Downstream task results.

Method	Open-domain QA				Fact Check.	Dial.
	NQ	TriviaQA	HotpotQA	ELI5	FEVER	WoW
	EM			R-L	ACC	F1
LLaMA-2-13b	19.74	68.71	15.64	19.46	73.23	13.90
Contriever	24.78	69.25	20.34	**20.71**	73.61	13.96
BM25	25.84	71.49	**27.23**	20.48	77.54	14.02
DPR⋆	**33.49**	**72.68**	23.13	**20.75**	75.27	14.17
MindRef (LLaMA-7b)	29.78	70.18	24.61	20.60	78.10	14.47
MindRef (LLaMA-13b)	29.68	71.60	25.48	20.24	**78.53**	14.33
MindRef (LLaMA-2-7b)	29.89	70.04	25.55	20.50	78.04	**14.48**
MindRef (LLaMA-2-13b)	**31.69**	**72.94**	**26.13**	20.61	**78.79**	**14.77**

3.2.3 Downstream Task Results

Downstream task results are presented in Table 3. MindRef, based on Llama-2-13b recalled passage, achieved the best scores of 72.94, 78.79, and 14.77 on the TriviaQA, FEVER, and WoW downstream tasks, respectively, validating the performance of LLM recall references in downstream tasks. On the open-domain question answering NQ dataset, although DPR performed excellently after full data training, MindRef also displayed highly competitive performance. On the other hand, in the TriviaQA and HotpotQA datasets, due to the length of the questions, BM25 achieved excellent performance by obtaining more vocabulary overlap, yet MindRef still achieved comparable or better performance in most cases. The unsupervised trained Contriever performed relatively poorly across all

Table 4: Ablation study results, with the left half showing the R-Precision at the coarse-grained page level and the right half showing Answer in Context for fine-grained passage.

Method	NQ	TriviaQA	HotpotQA	NQ	TriviaQA	HotpotQA
	R-Precision			Answer in Context		
MindRef	57.77	54.41	48.70	40.82	68.20	30.04
w/o weight	51.22	49.23	48.70	39.06	66.86	28.88
w/o SPRL	55.30	51.50	48.70	37.43	64.64	26.18
w/o first stage	32.22	24.87	23.36	36.27	63.33	24.16

tasks, emphasizing the crucial role of supervised training in enhancing the performance of dense retrieval models.

3.3 Ablation Study

In this subsection, we compare methods without weighted scores (w/o weight), without Short Prefix Recall Location (w/o SPRL), and without the first stage of document title recall (w/o first stage). The results are shown in Table 4.

Without weighted scores, relying solely on the recall scores from the second stage leads to a simultaneous decrease in performance for both coarse and fine-grained results, emphasizing the importance of considering scores from both stages. The model, by taking into account title scores, is more capable of selecting the correct document, and within the correct document, it is more likely to choose the correct reference. More results on the choice of weighted α can be found in Figure 4.

Without SPRL, recalling longer segments has a minor impact on page-level performance. However, it significantly affects the quality of fine-grained reference passage, where longer recall lengths paradoxically lead to decreased performance. This result is somewhat counterintuitive and might be due to all document knowledge being stored in the parameters during the pre-training phase, with a short prefix sufficient to locate the required reference. Longer references introduce redundancy and noise, thus lowering effectiveness. Notably, when using LLaMA-2-13b for recall, recalling complete passages on the NQ dataset takes about 600 minutes, while recalling short prefixes only requires 150 minutes, significantly reducing time costs. However, considering that dense retrieval takes about 20 minutes, further optimization of speed remains crucial. More experiments on prefix length can be found in Figure 5.

Without the first stage of document title recall, the quality of reference further declines, significantly impacting the quality of page retrieval. This indicates that using LLMs to directly recall references across a vast number of documents has considerable limitations and opportunities for improvement. The ability of merely prompting LLMs to recall and locate fine-grained reference passage is very limited, making the first stage of recalling document title identifiers crucial.

4 Related Work

Traditional retrieval methods rely on sparse (TF-IDF, BM25 (Robertson and Zaragoza, 2009)) or dense (ORQA (Lee et al., 2019), DPR (Karpukhin et al., 2020)) representations. However, dual-encoder dense retrieval faces limitations due to shallow interactions between independently encoded question and passage representations (Khattab and Zaharia, 2010).

Recent work explores using LLMs to generate identifiers for retrieval, aiming to simplify the process and enhance interaction compared to dual-encoder models. These approaches target page titles (De Cao et al., 2021), hierarchical paths (Tay et al., 2022), n-grams (Bevilacqua et al., 2022), multi-hop paths (Lee et al., 2022), multiple identifiers (Li et al., 2023b), or a two-stage approach with passages and URLs (Ren et al., 2023; Yue et al., 2025). These methods, however, predominantly retrieve predefined text segments, hindering flexible retrieval from arbitrary positions within full documents.

Leveraging LLMs to directly generate relevant knowledge (Fang et al., 2022) or augmenting models with LLM-generated context (e.g., GenRead (Yu et al., 2023), A+B (Tang et al., 2024)) has also shown promise for knowledge-intensive tasks. However, hallucination remains a significant challenge (Li et al., 2023a), potentially providing unreliable or fabricated information.

5 Conclusion

This paper introduces **MindRef**, a framework utilizing LLMs to independently recall reference passages for knowledge-sensitive tasks. Mimicking human information-seeking behavior, the LLM first recalls relevant document pages, and then locates specific passages within them. Beam search, constrained by Trie and FM-index structures, ensures that recalled content is a subset of existing texts. This framework is adaptable to various open-source LLMs, broadening their potential applications.

Limitations

Although MindRef demonstrates the potential of LLMs to recall reference passage in knowledge-sensitive tasks like humans, its application still faces several limitations. Firstly, this framework struggles to surpass the performance of the current SOTA retrieval models, especially those models that have been fine-tuned on specific tasks through supervision. In the future, there is a need to explore more effective ways of instruction tuning for recalling under constraints. At the same time, MindRef relies on document title identifiers for phased recall, meaning that its recall capability may be limited for documents lacking clear titles or identifiers.

Moreover, the framework finds it challenging to effectively recall documents that appear less frequently in the pre-training stage. This indicates that if a document appears infrequently in the training data of the LLM, or if the document content significantly differs from the training data, MindRef may encounter difficulties in recalling these documents. For the updating of documents and the injection of new knowledge, MindRef requires additional training to inject this new information into the model parameters. There is still a need to explore more efficient, lightweight methods for injecting new documents in the future.

Ethical Considerations

Our framework ensures that the generated content is entirely derived from reference materials, with Wikipedia as an example in this paper, thus not introducing additional significant ethical issues. However, in practical applications, we must ensure that the source document set relied upon is harmless to prevent the spread of inaccurate or harmful information.

References

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. *Advances in Neural Information Processing Systems*, 35:31668–31683.

Michael Burrows, D J Wheeler D I G I T A L, Robert W. Taylor, David J. Wheeler, and David Wheeler. 1994. A block-sorting lossless data compression algorithm.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Thomas H Cormen, Charles E Leiserson, et al. 2022. *Introduction to algorithms*. MIT press.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: long form question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3558–3567. Association for Computational Linguistics.

Yuwei Fang, Shuohang Wang, Yichong Xu, Ruochen Xu, Siqi Sun, Chenguang Zhu, and Michael Zeng. 2022. Leveraging knowledge in multilingual commonsense reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3237–3246. Association for Computational Linguistics.

Paolo Ferragina and Giovanni Manzini. 2000. Opportunistic data structures with applications. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 390–398. IEEE Computer Society.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 874–880. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided supervision for OpenQA with ColBERT. *Transactions of the Association for Computational Linguistics*, 9:929–944.

Omar Khattab and Matei Zaharia. 2010. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Hyunji Lee, Sohee Yang, Hanseok Oh, and Minjoon Seo. 2022. Generative multi-hop retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 1417–1436. Association for Computational Linguistics.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6086–6096. Association for Computational Linguistics.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6449–6464. Association for Computational Linguistics.

Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023b. Multiview identifiers enhanced generative retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6636–6648. Association for Computational Linguistics.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Ruiyang Ren, Wayne Xin Zhao, Jing Liu, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. TOME: A two-stage approach for model-based retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6102–6114. Association for Computational Linguistics.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Wei Tang, Yixin Cao, Jiahao Ying, Bo Wang, Yuyue Zhao, Yong Liao, and Pengyuan Zhou. 2024. A+ b: A general generator-reader framework for optimizing llms to unleash synergy potential. *arXiv preprint arXiv:2406.03963*.

Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Prakash Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. 2022. Transformer memory as a differentiable search index. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 809–819. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, et al. 2023b. LLaMA 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

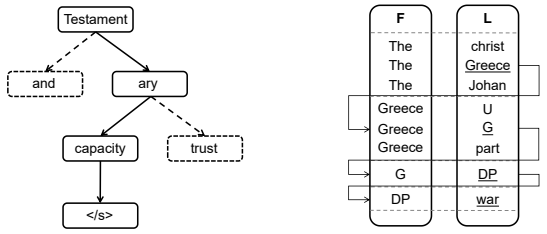
Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Chongjian Yue, Xinrun Xu, Xiaojun Ma, Lun Du, Zhiming Ding, Shi Han, Dongmei Zhang, and Qi Zhang. 2025. Extract information from hybrid long documents leveraging llms: A framework and dataset. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

A Preliminary

A.1 Trie



(a) Title Generation with Prefix Tree.

(b) Passage Prefix Generation with FM-index.

Figure 3: Constrained Decoding Methods: (a) Shows the process of an LLM generating title identifiers using a prefix tree. (b) Shows the process of an LLM generating passage prefixes in a document set via FM-index.

The Trie (Cormen et al., 2022), also known as a dictionary tree or prefix tree, is a tree-like data structure used to store an associative array where the keys are usually strings. Unlike a binary search tree, keys in a Trie are not stored directly within the nodes; instead, they are determined by the node's position in the tree. All descendants of a node have the same prefix, associated with the string corresponding to that node.

The overall process during constrained decoding using a Trie is shown in Figure 3a. Taking the generation of the title "Testamentary Capacity" as an example, the LLM first selects "Testament" from the set of token strings that start all titles. Subsequently, we can obtain the set of token strings {and, ary} following the string "Testament". After the LLM selects "ary", we get the prefix string "Testamentary", and finally continue to select new strings from the next set of token strings until the end-of-sequence token </s>is encountered, ceasing generation.

A.2 FM-Index

The FM-index (Ferragina and Manzini, 2000) is a data structure used for text retrieval that can store text efficiently with linear space complexity and support fast substring search operations. It is constructed based on the Burrows-Wheeler Transform (BWT) (Burrows et al., 1994). BWT is a method that converts a string into a form that is easy to compress. Given a string, BWT produces a transformed string through the following steps: generate all cyclic shifts of the string, sort all these shifts lexicographically, take the last character of each sorted shifted string to form a new string, which is the BWT result. For example, for the string "CABAC", the process of building the FM-index is as follows:

F					**L**
$\$^6$	C	A	B	A	C^5
A^2	B	A	C	$\$$	C^1
A^4	C	$\$$	C	A	B^3
B^3	A	C	$\$$	C	A^2
C^5	$\$$	C	A	B	A^4
C^1	A	B	A	C	$\$^6$

where $ is a special string termination token, the numbers in the upper right corner of the letters in the F and L columns are the corresponding position index numbers. The FM-index explicitly stores two main parts: the F column and the L column. The F column is the lexicographically sorted characters of the transformed string, and the L column is the result of BWT. In addition, it stores additional position information to recover the original string from the BWT result. When we want to query a substring, the FM-index starts from the last character of the substring, using the information in the F column and the L column to gradually narrow down the possible position range until the exact position of the substring is determined or the substring is determined to be non-existent.

The overall process during constrained decoding using FM-index is shown in Figure 3b. Considering the generated prefix "The Greece GDP warrants are not technically bonds as investors do" for example, it first starts from the string "The" generated from all corpus, and gets its corresponding L column string set {christ, Greece, Johan}. After "Greece" is selected by the LLM, we can get the next set {U, G, part}, and continue the iteration until reaching the set maximum prefix length to stop generating.

B Additional Details for Experiments

B.1 Datasets

We conduct extensive experiments on 6 knowledge-sensitive tasks from the KILT benchmark (Petroni et al., 2021). These tasks include open-domain QA tasks such as NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018), and ELI5 (Fan et al., 2019), the fact-checking task FEVER (Thorne et al., 2018), and the open-domain dialogue system WoW (Dinan et al., 2019). All experiments are tested using the public validation set as divided in the official KILT. Additional details of the datasets are presented in Table

Dataset	Task	Input Format	Output Format	Size
NQ (Kwiatkowski et al., 2019)	Open-domain QA	Question	Extractive	2837
HotpotQA (Yang et al., 2018)	Open-domain QA	Question	Short Abstractive	5600
TriviaQA (Joshi et al., 2017)	Open-domain QA	Question	Extractive	5359
ELI5 (Fan et al., 2019)	Open-domain QA	Question	Long Abstractive	1507
FEVER (Thorne et al., 2018)	Fact Checking	Claim	Classification	10444
WoW (Dinan et al., 2019)	Dialogue	Conversation	Long Abstractive	3054

Table 5: Additional details of the datasets.

5. All the data used in this paper come from the KILT benchmark(Petroni et al., 2021), and KILT is MIT licensed[4]. We evaluate the quality of coarse-grained pages and fine-grained reference passage, as well as the enhancement of reference for downstream tasks.

B.2 Prompt

In this subsection, we introduce the prompt used in the first stage for recalling coarse-grained title identifiers, in the second stage for recalling fine-grained reference passage, and in downstream tasks.

B.2.1 Prompt for the First Stage

- Open-domain QA: "Question: {}\n \n The Wikipedia article corresponding to the above question is:\n \n Title:"
- Fact Verification: "Claim: {}\n \n The Wikipedia article corresponding to the above claim is:\n \n Title:"
- Open-domain Dialogue System: "Conversation: {}\n \n The Wikipedia article corresponding to the above conversation is:\n \n Title:"

B.2.2 Prompt for the Second Stage

- Open-domain QA: "Question: {}\n \n The Wikipedia paragraph to answer the above question is:\n \n Answer:"
- Fact Verification: "Claim: {}\n \n The Wikipedia paragraph to support or refute the above claim is:\n \n Answer:"
- Open-domain Dialogue System: "Conversation: {}\n \n The Wikipedia paragraph to answer the above conversation is:\n \n Answer:"

[4]https://opensource.org/licenses/MIT

B.2.3 Prompt for Reading Comprehension

- Open-domain QA (NQ, TriviaQA, HotpotQA): "Refer to the passage below and answer the following question with just a few words.\n Passage: {}\n Q: {}\n A: The answer is"
- Open-domain QA (ELI5): "Refer to the passage below and answer the following question in detail.\n Passage: {}\n Q: {}\n A:"
- Fact Verification: "background: {}\n claim: {}\n Q: Is the claim true or false?\n A:"
- Open-domain Dialogue System: "background: {}\n {}\n "

B.3 Further Analysis

Different Values of Alpha In Figure 4, we compare the experimental results of MindRef when implemented based on Llama-2-13b with different α values. When $\alpha = 0.0$, it's equivalent to not having a two-stage weighted method, relying only on the scores from the second stage's fine-grained passage recall, resulting in the selection of suboptimal reference. With the increase of α, the model sees improvements in both page-level and passage-level results, proving the importance of the first stage document scores for the final reference selection. However, when α reaches 0.95 and continues to increase, the final performance actually decreases to some extent, indicating the need to find a balance between the two for better results.

Different Prefix Lengths In Figure 5, we conduct experiments with MindRef recalling different numbers of prefix tokens based on Llama-2-13b. We observe that longer prefix lengths do not bring additional performance improvements; on the contrary, they lead to a decrease in performance. Existing LLMs still perform better when generating shorter passages under constraints; longer passages

Method	NQ	TriviaQA	HotpotQA	NQ	TriviaQA	HotpotQA
	R-Precision			Answer in Context		
MindRef (LLaMA-7b)	54.46	57.03	44.56	36.87	58.48	25.55
MindRef (Vicuna-1.3-7b)	48.47	47.99	40.79	35.28	56.41	23.75
MindRef (LLaMA-13b)	54.42	55.53	46.30	37.72	60.96	26.34
MindRef (Vicuna-1.3-13b)	52.73	46.61	43.41	36.55	67.29	26.63
MindRef (LLaMA-2-7b)	56.33	56.43	46.20	38.07	62.88	27.55
MindRef (LLaMA-2-chat-7b)	3.31	1.12	0.98	4.09	3.97	3.43
MindRef (Vicuna-1.5-7b)	50.76	51.73	41.23	34.16	55.98	24.43
MindRef (LLaMA-2-13b)	57.77	54.41	48.70	40.82	68.20	30.04
MindRef (LLaMA-2-chat-13b)	1.94	1.60	1.55	6.38	7.93	4.71
MindRef (Vicuna-1.5-13b)	52.24	56.34	45.90	37.22	63.24	27.14

Table 6: On the NQ, TriviaQA, and HotpotQA datasets, experimental results after general fine-tuning of the model are presented. The left side shows the page-level R-Precision, while the right side displays the passage-level Answer in Context.

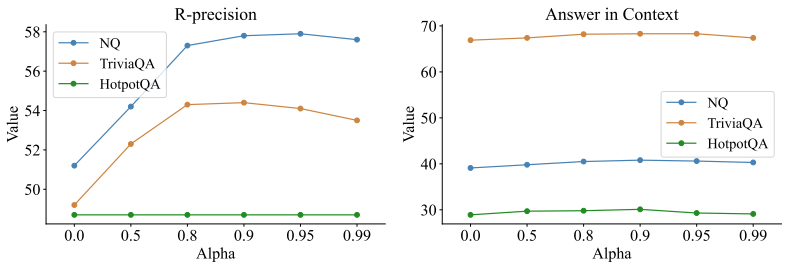


Figure 4: On the NQ, TriviaQA, and HotpotQA datasets, the page-level and passage-level experimental results for LLaMA-2-13b when setting α to {0.0,0.5,0.8,0.9,0.95,0.99}.

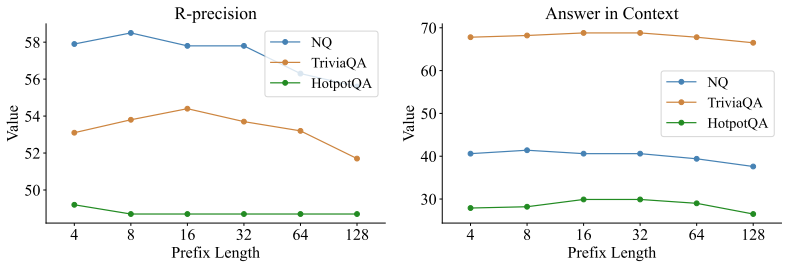


Figure 5: On the NQ, TriviaQA, and HotpotQA datasets, the page-level and passage-level experimental results for LLaMA-2-13b with different prefix token lengths l_{p_s} set to {4,8,16,32,64,128}.

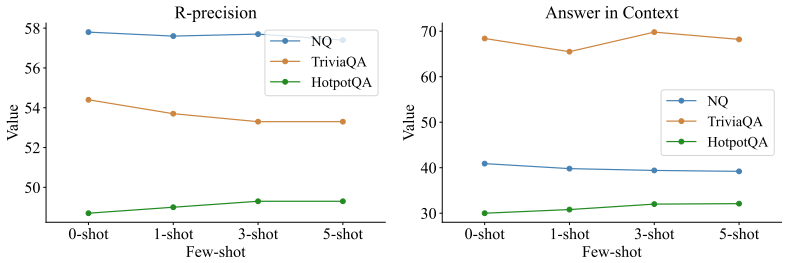


Figure 6: On the NQ, TriviaQA, and HotpotQA datasets, the page-level and passage-level experimental results for LLaMA-2-13b under {0,1,3,5}-shot few-shot prompt.

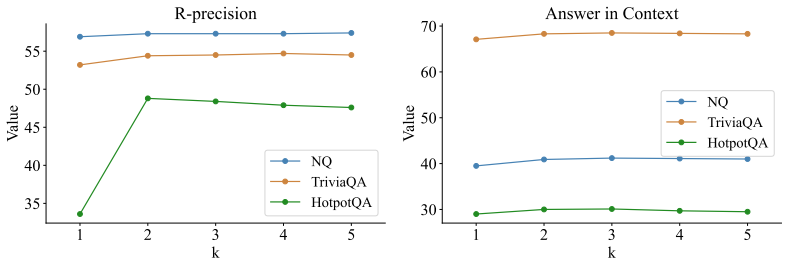


Figure 7: On the NQ, TriviaQA, and HotpotQA datasets, the page-level and passage-level experimental results for LLaMA-2-13b with the number of documents selected in the first stage k set to {1,2,3,4,5}.

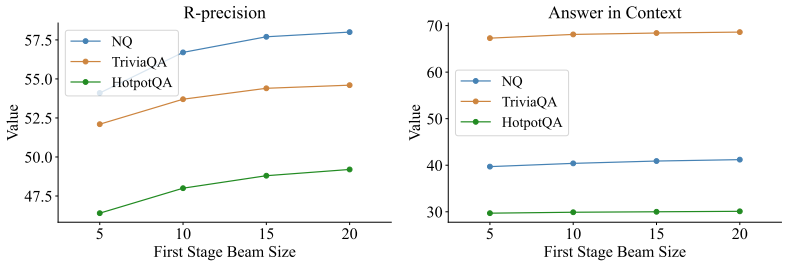


Figure 8: On the NQ, TriviaQA, and HotpotQA datasets, the page-level and passage-level experimental results for LLaMA-2-13b with different beam search sizes {4,8,16,32,64,128} set for the first stage.

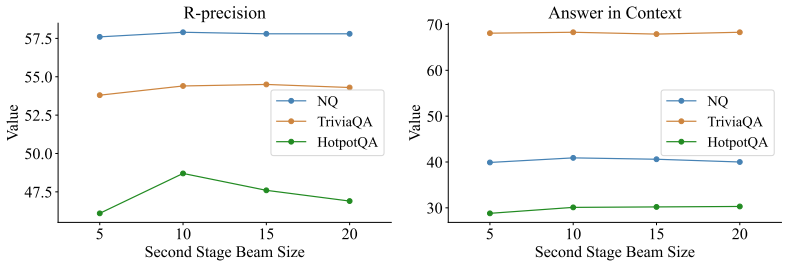


Figure 9: On the NQ, TriviaQA, and HotpotQA datasets, the page-level and passage-level experimental results for LLaMA-2-13b with different beam search sizes {4,8,16,32,64,128} set for the second stage.

introduce additional noise, resulting in decreased performance. However, overly short prefixes might also lack sufficient information, leading to an inability to accurately select the desired passage as a reference.

After General Fine-tuning of LLMs. We also test the Vicuna model (Chiang et al., 2023) and the LLaMA-2-chat model refined through reinforcement learning from human feedback (Touvron et al., 2023b), both of which underwent general fine-tuning. This general fine-tuning did not significantly enhance the performance of LLMs in recalling and locating reference. This may be due to the paradigm difference between the fine-tuning data and the recall location task, coupled with the fact that most knowledge was already acquired during the pre-training phase. By creating more diverse recall instruction tuning data, further improvements in model performance might be achieved. Detailed results can be found in Table 6.

Impact of Few-Shot. We explore adding few-shot prompt in the second stage of fine-grained recall and observed its impact on overall performance. This approach brought slight improvements only in the HotpotQA dataset, while showing a slight decline in NQ and TriviaQA. Importantly, adding more few-shot examples significantly reduced generation speed. This suggests that, although few-shot prompting offers a potential path for improvement, extensive exploration is still needed to devise more effective prompting methods. Detailed results can be found in Figure 6.

Impact of Document Selection (k) of First Stage Documents. In Figure 7, we conduct experiments to compare the effect of selecting different numbers of first-stage documents (denoted as k) in MindRef when implemented based on LLaMA-2-13b. We observe that the impact of k on the final

performance is not significant, as the necessary effective reference passages are usually contained within the first few documents. The suboptimal performance observed on the HotpotQA dataset when $k = 1$ can be attributed to the dataset requiring two documents to calculate R-Precision.

Impact of Beam Search Sizes. Figures 8 and 9 show the impact of setting different beam sizes in the first and second stages, respectively, in MindRef when implemented based on LLaMA-2-13b. For the first stage of recalling title identifiers, a larger beam size can achieve better page-level results, thereby slightly improving the effectiveness of the second stage of passage recall. However, in the second stage of fine-grained passage recall, the improvement brought by a larger beam size is not significant and may even lead to a slight decline, possibly due to the introduction of additional noise by a larger beam size.

Memory Usage Analysis. Dense retrieval methods such as Contriever and DPR require over 60GB of memory usage. In contrast, sparse retrieval methods use far less memory, only needing 17GB. The MindRef framework, utilizing FM-index and Trie indexing, requires only 8GB when pre-encoding and storing all documents with FM-index, and the Trie storing all title identifiers needs just 25MB, which is negligible. Compared to sparse and dense retrieval methods, the recall framework effectively saves memory.

B.4 Case Study

In Supplementary Material Tables 7 to 12, we present reference cases obtained using the Gold Standard, BM25, and the MindRef framework with LLaMA-2-13b on the NQ, TriviaQA, and HotpotQA datasets. By generating passages more aligned with the question, MindRef achieves results that contain the answer in Supplementary Material Tables 7, 9, and 11. Supplementary Material Table 8 showcases a biology question; although the passage recalled and located by MindRef does not contain the annotated answer, it provides a more detailed description of the location and process of pancreatic enzyme cleavage of peptide bonds. However, Supplementary Material Tables 10 and 12 show instances where MindRef's recall failed. This is because merely generating a relevant prefix sometimes cannot ensure that the subsequent part will definitely contain the answer, leading to passages that are only broadly related. Ensuring the flexibility of recall and location while considering more subsequent information still requires further exploration. Nevertheless, we can also note that the references obtained by LLM recall are more natural and easier to read compared to those with predefined segmented beginnings. Finally, compared to the NQ dataset, questions in the TriviaQA and HotpotQA datasets are longer, allowing for more vocabulary overlap between questions and passages, which enhances the effectiveness of BM25 sparse retrieval.

Query: what is the legal age for marriage in australia
Gold Standard: Part II (s.10-21) deals with the marriageable age and the marriage of minors. In the original 1961 Act, marriageable age was set at 16 for females and 18 for males. However, under section 12 of the original 1961 Act a female 14 or 15 years or a male 16 or 17 years could apply to the court for permission to marry. The marriageable age was equalised in 1991 by the "Sex Discrimination Amendment Act 1991", which raised the marriageable age of females to 18 .
BM25: marriage. Before 2004, there was no definition of marriage in the 1961 Act, and instead the common law definition used in the English case "Hyde v Hyde" (1866) was considered supreme. Though s.46(1) of the Act required celebrants to explain the legal nature of marriage in Australia to a couple as "the union of a man and a woman to the exclusion of all others, voluntarily entered into for life", these words were descriptive or explanatory, rather than outlining what constituted a legally valid marriage in
MindRef: Marriage in Australia is regulated by the federal "Marriage Act 1961" (Cth) , which applies uniformly throughout Australia (including its external territories) to the exclusion of all state laws on the subject. Australian law recognises only monogamous marriages, being marriages of two people, including same-sex marriages, and does not recognise any other forms of union, including traditional Aboriginal marriages, polygamous marriages or concubinage. The marriage age for marriage in Australia is 18 years, but in "unusual and exceptional circumstances" a person aged 16 or 17 can marry with parental consent

Table 7: In the NQ dataset, the Gold Standard, BM25, and MindRef reference passage for the query "what is the legal age for marriage in Australia" are provided. The parts containing the answer are highlighted with a grey background.

Query: where does cleavage of the peptide bond by chymotrypsin occur
Gold Standard: "In vivo", chymotrypsin is a proteolytic enzyme (serine protease) acting in the digestive systems of many organisms. It facilitates the cleavage of peptide bonds by a hydrolysis reaction, which despite being thermodynamically favorable, occurs extremely slowly in the absence of a catalyst. The main substrates of chymotrypsin are peptide bonds in which the amino acid N-terminal to the bond is a tryptophan, tyrosine, phenylalanine, or leucine. Like many proteases, chymotrypsin also hydrolyses amide bonds "in vitro", a virtue that enabled the use of substrate analogs such as N-acetyl-L-phenylalanine p-nitrophenyl amide for enzyme assays.
BM25: 149, producing α-chymotrypsin (which is more active and stable than π-chymotrypsin). The resulting molecule is a three-polypeptide molecule interconnected via disulfide bonds. "In vivo", chymotrypsin is a proteolytic enzyme (serine protease) acting in the digestive systems of many organisms. It facilitates the cleavage of peptide bonds by a hydrolysis reaction, which despite being thermodynamically favorable, occurs extremely slowly in the absence of a catalyst. The main substrates of chymotrypsin are peptide bonds in which the amino acid N-terminal
MindRef: Chymotrypsin cleaves peptide bonds by attacking the unreactive carbonyl group with a powerful nucleophile, the serine 195 residue located in the active site of the enzyme, which briefly becomes covalently bonded to the substrate, forming an enzyme-substrate intermediate. Along with histidine 57 and aspartic acid 102, this serine residue constitutes the catalytic triad of the active site. These findings rely on inhibition assays and the study of the kinetics of cleavage of the aforementioned substrate, exploiting the fact that the en

Table 8: In the NQ dataset, the Gold Standard, BM25, and MindRef reference passage for the query "where does cleavage of the peptide bond by chymotrypsin occur" are provided. The parts containing the answer are highlighted with a grey background.

Query: Which expression is associated with the sinking of the HMS Birkenhead at Gansbaai near Cape Town, South Africa, in Febuary 1852?
Gold Standard: Only 193 of the estimated 643 people on board survived, and the soldiers' chivalry gave rise to the unofficial "women and children first" protocol when abandoning ship, while the "Birkenhead drill" of Rudyard Kipling's poem came to describe courage in face of hopeless circumstances.
BM25: HMS "Birkenhead, also referred to as HM Troopship "Birkenhead or Steam Frigate "Birkenhead", was one of the first iron-hulled ships built for the Royal Navy. She was designed as a steam frigate, but was converted to a troopship before being commissioned. She was wrecked on 26 February 1852, while transporting troops to Algoa Bay at Danger Point near Gansbaai, 87 miles (140 kilometres) from Cape Town in the Cape Colony. There were not enough serviceable lifeboats for all the passengers, and the soldiers
MindRef: The sinking of the "Birkenhead" is one of the earliest maritime disaster evacuations during which the concept of "women and children first" is known to have been applied."Women and children first" subsequently became standard procedure in relation to the evacuation of sinking ships, in fiction and in life. The term "Birkenhead drill" became defined as courageous behaviour in hopeless circumstances and appeared in Rudyard Kipling's 1893 tribute to the Royal Marines, "Soldier an' Sailor Too": To take your chance in the thick of a rush, with firing all about, Is nothing so bad when you've

Table 9: In the TriviaQA dataset, the Gold Standard, BM25, and MindRef reference passage for the query "Which expression is associated with the sinking of the HMS Birkenhead at Gansbaai near Cape Town, South Africa, in Febuary 1852?" are provided. The parts containing the answer are highlighted with a grey background.

Query: Who was the first winner of 'I'm A Celebrity, Get Me Out Of Here'?
Gold Standard: The first series of I'm a Celebrity...Get Me Out of Here! was broadcast on ITV from 25 August to 8 September 2002. Ant & Dec presented the main show on ITV, whilst Louise Loughman hosted the spin-off show "I'm a Celebrity...Get Me Out of Here! NOW!" on ITV2. The winner of this series was radio DJ Tony Blackburn.
BM25: The first series of I'm a Celebrity...Get Me Out of Here! was broadcast on ITV from 25 August to 8 September 2002. Ant & Dec presented the main show on ITV, whilst Louise Loughman hosted the spin-off show "I'm a Celebrity...Get Me Out of Here! NOW!" on ITV2. The winner of this series was radio DJ Tony Blackburn. The show began with 8 celebrity contestants. The contestants take part in daily trials to earn food All ratings are taken from the UK
MindRef: I'm a Celebrity...Get Me Out of Here! is a British reality TV series in which a number of celebrities live together in a jungle environment for a number of weeks, competing to be crowned "King" or "Queen of the Jungle". The show was originally created in the United Kingdom by the factual programmes department of ITV's then London franchise, London Weekend Television and developed by a team including James Allen, Natalka Znak, Brent Baker and Stewart Morris. The first episode aired on 25 August 2002 . It is now produced by ITV Studios and has been licensed globally to countries including the United States

Table 10: In the TriviaQA dataset, the Gold Standard, BM25, and MindRef reference passage for the query "Who was the first winner of 'I'm A Celebrity, Get Me Out Of Here'?" are provided. The parts containing the answer are highlighted with a grey background.

Query: 2014 S/S is the debut album of a South Korean boy group that was formed by who?
Gold Standard: 2014 S/S is the debut album of South Korean group WINNER. It was released on August 12, 2014 by the group's record label, YG Entertainment. The members were credited for writing the lyrics and composing the majority of the album's songs.
BM25: S is a South Korean project group consisting of three members: Kangta, Lee Ji-hoon and Shin Hye-sung. The group debuted in 2003, under the SM Entertainment label. After 11 years, they released and promoted another mini-album in 2014. In 2003, S released their first album "Fr.In.Cl", which stands for "Friends in Classic". In 2014, after more than a decade in hiatus, the group released their second mini-album "Autumn
MindRef: 2014 S/S is the debut album of South Korean group WINNER. It was released on August 12, 2014 by the group's record label, YG Entertainment. The members were credited for writing the lyrics and composing the majority of the album's songs. The members produced the majority of the music for this album themselves, with the help of other producers such as Choice 37, B.I., Airplay, and others. The album was highlighted for incorporating elements generally absent from K-pop releases, including hints of acoustic

Table 11: In the HotpotQA dataset, the Gold Standard, BM25, and MindRef reference passage for the query "2014 S/S is the debut album of a South Korean boy group that was formed by who?" are provided. The parts containing the answer are highlighted with a grey background.

Query: Who is the fictional head of a British Secret Service division and for which a one-time missionary was the inspiration for?
Gold Standard: Charles Fraser-Smith (26 January 1904 – 9 November 1992) was an author and one-time missionary who is widely credited as being the inspiration for Ian Fleming's James Bond quartermaster Q. During World War II, Fraser-Smith worked for the Ministry of Supply, fabricating equipment nicknamed "Q-devices" (after Q-ships) for SOE agents operating in occupied Europe. Prior to the war, Fraser-Smith had worked as a missionary in North Africa. After the war he purchased a dairy farm in Bratton Fleming, Devon, where he died in 1992.
BM25: Q is a fictional character in the James Bond films and film novelisations. Q (standing for Quartermaster), like M, is a job title rather than a name. He is the head of Q Branch (or later Q Division), the fictional research and development division of the British Secret Service. The use of letters as pseudonyms for senior officers in the British Secret Intelligence Service was started by its first director Captain Sir Mansfield George Smith-Cumming (1859-1923) who signed himself with a C written in
MindRef: Ian Fleming created the fictional character of James Bond as the central figure for his works. Bond is an intelligence officer in the Secret Intelligence Service, commonly known as MI6. Bond is known by his code number, 007, and was a Royal Naval Reserve Commander. Fleming based his fictional creation on a number of individuals he came across during his time in the Naval Intelligence Division and 30 Assault Unit during the Second World War, admitting that Bond "was a compound of all the secret agents and commando types I met during the war". Among those types were his brother, Peter, who had been involved in behind-the-lines operations in Norway and Greece during the war.

Table 12: In the HotpotQA dataset, the Gold Standard, BM25, and MindRef reference passage for the query "Who is the fictional head of a British Secret Service division and for which a one-time missionary was the inspiration for?" are provided. The parts containing the answer are highlighted with a grey background.

LLMs syntactically adapt their language use to their conversational partner

Florian Kandra, Vera Demberg, Alexander Koller
Department of Language Science and Technology
Saarland University, Saarbrücken, Germany
flka00003@uni-saarland.de
{vera, koller}@coli.uni-saarland.de

Abstract

It has been frequently observed that human speakers align their language use with each other during conversations. In this paper, we study empirically whether large language models (LLMs) exhibit the same behavior of conversational adaptation. We construct a corpus of conversations between LLMs and find that two LLM agents end up making more similar syntactic choices as conversations go on, confirming that modern LLMs adapt their language use to their conversational partners in at least a rudimentary way.

1 Introduction

It has been documented broadly that when humans talk to each other, they adapt their language use to their communication partners by coordinating their behavior and language. Humans *align* not only their gestures, posture, and speech rate (Holler and Wilkin, 2011; Shockley et al., 2009; Jungers and Hupp, 2009), but also their linguistic decisions at deeper levels, such as semantics and syntax (Bock, 1986; Garrod and Anderson, 1987). In other words, the distribution over syntactic structures of two human speakers becomes more similar as a conversation progresses.

In this paper, we investigate whether large language models (LLMs) adapt their syntactic choices to their conversational partners as well. While it is well known that LLMs can be explicitly prompted towards embodying different "personas" and changing the style of the language they generate (Deshpande et al., 2023; Thillainathan and Koller, 2025), it is unclear whether merely being present in a conversation with an interlocutor is sufficient to make LLMs adapt their language use to their interlocutor's. The ability to adapt to the communication partner's language is associated with increased success in goal-oriented conversations (Reitter and Moore, 2014), and it enables a dialogue system to meet a user's language use rather than requiring the user to adapt to the system (Schlangen, 2022). Language models will only serve as effective foundations for dialogue systems if they prove capable of implicitly adapting to a user's language.

To this end, we create a new dataset of conversations between LLMs in which both LLMs are prompted to initially exhibit different language use. We then measure the dynamics of syntactic language adaptation over the course of the conversations, using a method adapted from the human-human analysis of Reitter and Moore (2014). We find that GPT-4o (Hurst et al., 2024) and Llama-3-8B (Grattafiori et al.) conversations show statistically significant adaptation when comparing syntactic repetitions within conversations against repetitions across conversations, replicating Reitter's findings for human conversations. We further show this is a continuous process active throughout conversations and conclude by discussing whether these findings demonstrate "human-like" alignment in LLMs.

Our code and data are publicly available[1].

2 Background

As we mentioned above, humans adapt their language use to their communication partners across various linguistic levels. In this paper, our focus is on *syntactic* adaptation: Do the distributions over the syntactic structures that two interlocutors produce become more similar over the course of a conversation?

In the psycholinguistics literature on human communication, two separate (but not exclusive) mechanisms have been proposed to explain the mutual adaptation of language use. Rasenberg et al. (2020) contrast two theoretical views that explain the process through *alignment* on different cognitive lev-

[1] https://github.com/coli-saar/llm-language-accomodation

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 873–886
July 27 - August 1, 2025

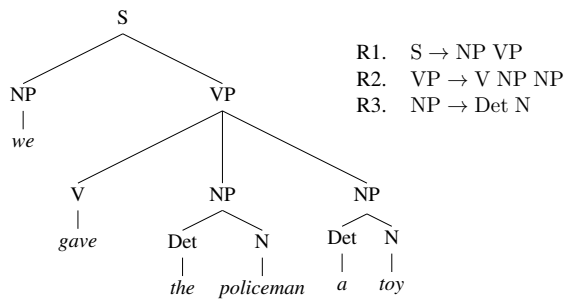


Figure 1: Phrase Structure Tree and Extracted Production Rules

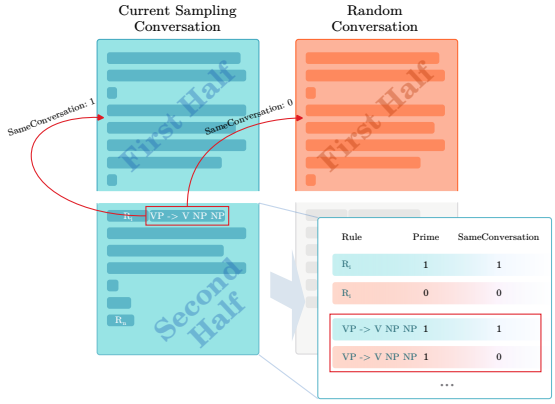


Figure 2: Sampling process to analyse syntactic alignment. Samples are drawn by checking rule occurences in the same conversation and in different random conversations.

els: on a conscious level, in which cooperative decisions establish a situational common ground (Brennan and Clark, 1996), and a subconscious level, in which automatic *priming* leads to aligned representational states (Pickering and Garrod, 2004). In psycholinguistics, priming refers to a process in which encountering a word or construction temporarily increases the activation of a cognitive representation, thereby increasing the probability for the word or construction to be reproduced.

In this paper, we study the conversational behavior of artificial, LLM-based agents. We will primarily focus on the level of outwardly observable changes to the language use and describe it with the theory-neutral word *adaptation*. We will discuss in Section 5 the extent to which concepts like alignment and priming can apply to LLMs.

Related Work. Cai et al. (2024) examined syntactic adaptation in LLMs, with a focus on short-term priming. They showed that ChatGPT and Vicuna are more likely to complete a sentence with a double object or a prepositional object when primed with a sentence of the respective type. We extend this research to long conversations with natural sentences rather than carefully constructed one-sentence stimuli.

3 Measuring human-human adaptation

The phenomenon of long-term syntactic adaptation was first measured on corpora of human-human dialogues by Reitter and Moore (2014). The basic idea is to determine whether the usage frequency of a syntactic structure (specifically, a rule in a context-free grammar) in the first half of a conversation has a statistical impact on its frequency in the second half.

We follow Reitter in splitting each conversation in a dialogue corpus into two parts. We call the first 49% of each conversation PRIME and the last 49% of each conversation TARGET; the middle 2% are discarded to ensure that we measure long-term adaptation as opposed to short-term priming. On corpora that are not already syntactically annotated, we parse each conversation with the Neural Berkeley Parser[2] (Kitaev and Klein, 2018; Kitaev et al., 2019), to obtain a set of context-free production rules for the PRIME and TARGET section of each conversation, respectively (see Fig. 1).

Adaptation takes place if rule repetitions are more likely between the PRIME and TARGET of the same conversation (where adaptation is possible), compared to a PRIME and TARGET of different conversations (where no adaptation could have taken place). In order to fit a model that can find such an effect, we sample data from our conversations in the following way: For each rule across the TARGETs of all conversations, we draw two samples, one for which we check whether that rule has been uttered within the same conversation, but by the other speaker, and one sample for which we check whether that rule has been uttered by a random speaker of a random other conversation. Figure 2 depicts this process. For every sample, we encode whether a rule was found this way in a binary variable *Prime*. Another binary variable, *SameConv*, is used to indicate whether we looked for a prime in the same conversation (1) or in a different, random conversation (0). If repetitions are more likely between speakers within conversations, such that we see an effect of *SameConv* on *Prime*, we take that as evidence of cross-speaker adaptation.

[2]We used the benepar_en3_large model of the benepar python package for parsing and spacy's en_core_web_md model for tokenization.

We further include features representing the log-frequency of rules across all conversations (*ln(Freq)*), as more frequent rules are expected to be more likely to appear in any PRIME, and a variable *ln(Size)*. This second variable encodes the amount of different rules that a speaker used in the PRIME of a conversation, i.e. the size of the set of rules that we use to look for a prime; a larger set increases the probability of any rule to occur. We follow Reitter and Moore (2014) in excluding rules that appear only once in the whole dataset and rules that have disproportionately high frequencies (around 0.3% of each dataset), because these rules are never primed or almost always primed. Including these rules in the analysis does not substantially change the results (see Appendix D). We further remove structures that are lexically identical.

Our analysis differs from Reitter's original method in two aspects. First, we consider only overlaps between rule uses in TARGET with uses in PRIME by the other speaker. This eliminates effects that solely stem from speaker idiosyncrasies or the conditioning of LLM-generated language on its own prior output. Second, our analysis includes the set size of rules used to check for a prime.

3.1 Alignment in human conversations

We replicate Reitter's results on human-human conversations to ensure that we obtain comparable results after our modifications. We use the method described above to analyze the Switchboard corpus (Marcus et al., 1994), which comprises 650 syntactically annotated telephone conversations (see Fig. 4 in Appendix B for an overview of its composition). This is in contrast to Reitter's work, which used the HCRC Map Task corpus (Anderson et al., 1991), consisting of task-oriented conversations. By looking at Switchboard as opposed to Map Task, we demonstrate alignment effects on non-task-oriented conversations, facilitating comparison with LLM-generated conversations, and we make use of hand-annotated rather than automatically parsed syntactic structures.

We fit a mixed-effects logistic regression to the sampled data using the generalized linear mixed models (GLMM) of Python's pymer4 (v0.8.2) package. We included a nested random intercept for conversations and speakers and a random slope for *ln(Freq)* and centered fixed effects except *SameConv*. We selected the model through a backward selection process. Results are shown in Table 1.

We find that *SameConv* ($\beta = 0.228, p < 0.001$) has a significant positive effect, replicating Reitter's findings that humans align syntactically to their partners over the course of a conversation.

4 Measuring LLM-LLM adaptation

We follow the same method to analyze syntactic adaptation in conversations of GPT-4o and Llama-3-8B.

Dataset. One challenge towards this goal is the availability of a suitable dataset of LLM conversations. We require a dataset consisting of long natural conversations (with no intervening task prompts) in which the speakers use varying syntactic structures to make adaptation possible and evenly distributed utterance lengths.

Existing datasets of conversations with LLMs do not satisfy these requirements. UltraChat (Ding et al., 2023) is a dataset of LLM-LLM conversations, but these conversations follow simple question-answering between a user and a model "persona". Conversations are too short and there is no variability between the language use across conversations. By contrast, available datasets of human-LLM conversations, such as WildChat (Zhao et al., 2024), consist of conversations that each have unique instructions by the user. This makes conversations incomparable and therefore unsuitable for a statistical analysis of adaptation.

We therefore created our own dataset by letting GPT-4o and Llama-3-8B[3] converse with themselves. We created 17 different conversational agents with identical system prompts, except for an initial specification of a "language persona" that is unique to each agent. We then generated conversations between pairs of LLM agents by iteratively prompting each of them for the next utterance, including the context of the conversation history. Iterations were stopped, once a conversation surpassed a predefined length threshold. All prompts for managing the conversations and defining the language personas can be found in Appendix A.

To ensure sufficient variety in the agents' language use, we further generated conversations where each agent conversed with itself. We then calculated how often each syntactic rule was used and normalized these frequencies to create a discrete probability distribution of syntactic rules for each agent. To compare these distributions, we measured their distances using the Jensen-Shannon

[3]We used GPT-4o-2024-08-06 and Meta-Llama-3-8B-Instruct with default parameters.

	Switchboard Corpus				GPT Corpus				Llama Corpus			
	β	SE	z	$p > \|z\|$	β	SE	z	$p > \|z\|$	β	SE	z	$p > \|z\|$
Intercept	-2.927	0.018	-158.8	0.000	-2.031	0.048	-42.5	0.000	-2.016	0.049	-41.0	0.000
ln(Freq)	1.174	0.008	143.2	0.000	1.275	0.028	45.6	0.000	1.333	0.030	44.5	0.000
SameConv	0.228	0.023	9.9	0.000	0.198	0.056	3.5	0.000	0.505	0.052	9.6	0.000
ln(Size)	1.402	0.033	41.9	0.000	1.175	0.107	11.0	0.000	0.825	0.147	5.6	0.000
ln(Freq):SameConv	-0.101	0.01	-9.8	0.000	-0.146	0.035	-4.2	0.000	-0.252	0.032	-7.8	0.000
ln(Freq):ln(Size)	0.068	0.015	4.7	0.004	0.266	0.062	4.3	0.000	-0.005	0.088	-0.06	0.952

Table 1: The regression models for the Switchboard corpus (left) and the GPT corpus (middle) and LLama corpus (right). Effects show high significance except for the interaction between *ln(Freq)* and *ln(Size)* in the Llama corpus.

divergence (JSD). See Figures 9 and 10 in Appendix C for details. The results confirm a high degree of syntactic variety, with JSD values of up to 0.69 for GPT-4o and 0.70 for Llama-3-8B.

Adaptation in LLM conversations. We generated 136 conversations for each model by pairing up every conversational agent with every other conversational agent, all on the topic "What makes a day a good day?" In GPT conversations, twelve of them ended in repeating patterns (see Appendix E) for Llama only one; we excluded them and used the remaining 124 and 135 conversations to form the GPT and Llama corpora respectively. The distributions of conversation and utterance lengths closely mirror that of the Switchboard corpus (cf. Fig. 5, Fig. 6 and Fig. 4 in Appendix B).

We ran the analysis described in Section 3.1 on the corpora. Because agents appear in multiple different conversations, we took care not to sample from identical agents from other conversations. Fixed effects, except *SameConv*, are centered. The models were selected using backward selection. The results are shown in Table 1.

For both Llama and GPT, *SameConv* has a significant positive effect on *Prime* ($\beta = 0.198$, $p < 0.001$ and $\beta = 0.364$, $p < 0.001$), showing that there is syntactic adaptation.

Fine-grained tracking of the adaptation process. To gain a deeper understanding of the adaptation process performed by the LLMs, we performed a fine-grained analysis of adaptation over the course of the conversation. To do so, we again directly compared the distributions of syntactic structures used by two different agents; however, this time, we focused on comparing the distributions to see how they evolve throughout a conversation. To obtain reliable estimates of the distributions, we created 520 conversations between agents 5 and 6, a pair of agents with moderate initial JSD (cf. Fig. 9, 10), while keeping the topic the same (cf. Appendix A). Due to repeating patterns, we excluded 14 conversations of GPT-4o and 7 of the Llama model.

To observe how the similarity of the two agents' distributions evolves, we split the remaining conversations into sections of 200 words (see Fig. 7, 8 in Appendix B for an overview of the data), and compare the distributions of used rules by the two agents for each split. Again, we obtain these probability distributions by normalizing rule frequencies for each split across conversations. To estimate the variance of these calculations, we perform them on 100 bootstraps of the data. Each bootstrap is a collection of conversations that is drawn with replacement from the original GPT and Llama conversations, therefore resulting in 100 collections of 506 and 513 randomly drawn conversations for each model. We report the means and standard deviations of these 100 JSD values across splits in Fig. 3.

We find that the mutual adaptation of the two LLM agents is a gradual process that persists throughout the conversation for both Llama-3-8B and GPT-4o. The rate of adaptation is relatively constant, with the strongest adaptation happening in the first split.

5 Discussion

Throughout the paper, we have avoided using the words "alignment" and "priming" for the LLMs' adaptation process to steer clear of any connotations about human cognitive processes. While we have established that the LLMs' syntax becomes increasingly similar to their conversational partner's, this does not necessarily mean that this process is driven by a similar underlying mechanism.

An LLM does not maintain an explicit mental model of its interlocutor's language use and does not make conscious decisions on coordinating it with its interlocutor. Thus it seems inappropriate, under the notion of alignment sketched in Section 2, to explain the LLMs' adaptive behavior as alignment. At the same time, priming effects in humans

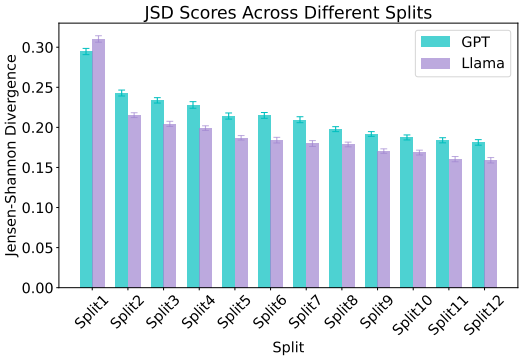


Figure 3: Jensen-Shannon divergence scores between agents 5 and 6 across splits of conversations.

are usually assumed to impact their language use only in the short term. One conceivable explanation for GPT-4o's and Llama-3-8B's ability to perform long-term adaptation is that they condition the language they produce on the previous conversation (a mechanism that is similar to priming in humans), but have a much larger capacity than humans for remembering the verbatim conversational context.

The findings of our second experiment also support this idea: A gradual adaptation that appears with increased context length underpins the intuition that LLMs can adapt to longer contexts, and that longer context correlates with an increased influence on the production of syntactic structures. Different from humans, short-term effects, like those reported in Cai et al. (2024), may therefore be driven by the same principles as long-term adaptation in LLMs. A more detailed analysis would be an interesting avenue of future research.

6 Conclusion

We showed that GPT-4o and Llama-3-8B can gradually adapt their language use to their conversational partner, to an extent that is similar to what we observe in human-human conversations. This observation goes beyond previous findings, which indicated that an LLM's language use can be controlled through explicit instructions and influenced by priming from the previous utterance. A more detailed comparison of the mechanisms that humans and LLMs use to achieve such long-term adaptation is an interesting avenue of future work.

7 Acknowledgements

This research was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), which supported the work of FK – Project KO 2916/3-1 – and VD – project number 389792660 – TRR 248 – CPEC, see https://perspicuous-computing.science.

Florian Kandra would like to thank Albert Kandra, who helped with extensive analytical conversations during the preparation of this paper.

8 Limitations

This work focuses on texts generated with GPT-4o and Llama-3-8B. We decided to use these models, as they cover a range between one of the highest performing accessible models and an open sourced, but smaller and less performant model. The findings suggest that syntactic adaptation generalizes to other models as well, which is supported by the findings of Cai et al. (2024). Further research could explore additional models to investigate the underlying factors that may influence syntactic adaptation.

Our study concentrates on syntactic structures of the English language. LLMs may exhibit similar behavior for other languages and other linguistic features, also of different modalities (e.g. intonation, speech rate). It would be interesting to investigate this in future work.

Furthermore, in this study we controlled for topicality by keeping the topic of all conversations identical. It is unclear whether topicality has an effect on syntactic structures, but there is evidence that lexical choices influence the syntax at least to some extent (lexical boost, Cai et al., 2024). To what extent topicality can have an effect on syntactic choices in LLMs is left as an avenue for future research.

Moreover, our study currently only focuses on LLM-LLM conversations. It would be interesting to see how these effects impact human-LLM conversations, especially given the societal impact of human-LLM interaction.

The analysis that we adapt from Reitter and Moore (2014) loses information by encoding the presence of syntactic structures in a binary variable. While the analysis is suitable for capturing adaptation in general, it lacks the sensitivity to account for the occurrence rate of rules in a meaningful way.

9 Ethical Considerations

We believe that our work is unlikely to have an immediate ethical or societal impact. However, there is potential that the reported effects serve as a footprint of LLM generated conversations – we didn't

prompt the model to adapt to the language, but this effect appears inherently. This potentially leads to patterns that are intrinsic to LLMs, which could be leveraged to detect LLM generated conversations.

References

Anne H. Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, Catherine Sotillo, Henry S. Thompson, and Regina Weinert. 1991. The hcrc map task corpus. *Language and Speech*, 34(4):351–366.

J.Kathryn Bock. 1986. Syntactic persistence in language production. *Cognitive Psychology*, 18(3):355–387.

S. E. Brennan and H. H. Clark. 1996. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22:1482–1493.

Zhenguang Cai, Xufeng Duan, David Haslett, Shuqi Wang, and Martin Pickering. 2024. Do large language models resemble humans in language use? In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 37–56, Bangkok, Thailand. Association for Computational Linguistics.

Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. 2023. Toxicity in chatgpt: Analyzing persona-assigned language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1236–1270, Singapore. Association for Computational Linguistics.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore. Association for Computational Linguistics.

Simon Garrod and Anthony Anderson. 1987. Saying what you mean in dialogue: A study in conceptual and semantic co-ordination. *Cognition*, 27(2):181–218.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Judith Holler and Katie Wilkin. 2011. Co-speech gesture mimicry in the process of collaborative referring during face-to-face dialogue. *Journal of Nonverbal Behavior*, 35(2):133–153.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, ..., and Yury Malkov. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Melissa K. Jungers and Julie M. Hupp. 2009. Speech priming: Evidence for rate persistence in unscripted speech. *Language and Cognitive Processes*, 24(4):611–624.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Martin J. Pickering and Simon Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27(2):169–190.

Marlou Rasenberg, Asli Özyürek, and Mark Dingemanse. 2020. Alignment in multimodal interaction: An integrative framework. *Cognitive Science*, 44(11):e12911.

David Reitter and Johanna D. Moore. 2014. Alignment and task success in spoken dialogue. *Journal of Memory and Language*, 76:29–46.

David Schlangen. 2022. Norm participation grounds language. In *Proceedings of (Dis)embodiment 2022: A CLASP Conference*.

Kevin Shockley, Daniel C. Richardson, and Rick Dale. 2009. Conversation and coordinative structures. *Topics in Cognitive Science*, 1(2):305–319.

Sarubi Thillainathan and Alexander Koller. 2025. Fine-grained controllable text generation through in-context learning with feedback. In *AAAI 2025 Workshop on AI for Education – Tools, Opportunities, and Risks in the Generative AI Era*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatgpt interaction logs in the wild. *Preprint*, arXiv:2405.01470.

A Prompts

A.1 System Prompt

The following template was used as the system prompt in the data generation process:

> You are in a conversation. There are two speakers, SpeakerA and SpeakerB.
> You are SpeakerA. The conversation will consists of turns in the form:
> [SpeakerA's utterances]
> [SpeakerB's utterances]
> [SpeakerA's utterances]
> ...
> You need to only give [SpeakerA's utterances]. You will be prompted by [Language] that will instruct you on the language that you shall use as SpeakerA. Further, you will be prompted by [Topic], the topic of the conversation. Behave as in a normal conversation with SpeakerB to discuss the [Topic]. [Language] {That agent's specific persona, see item A.2)}. [Topic] What makes a day a good day?

A.2 Language Personas

The following language personas were used to vary the language of each agent. Language personas are inserted into the system prompt at the designated position.

1. Your language is precise, and unambiguous. You use clear and simple sentences.
2. Your language is gentle and thoughtful. You use concise and not overly complex sentences, to convey meaning efficiently.
3. Your language is dynamic, and provocative. You often use vivid metaphors.
4. Your language is introspective, and deliberate. You use contemplative phrasing.
5. Your language is smooth and reassuring. You employ gentle pauses and a steady rhythm.
6. Your language is analytical and precise. You use complex sentence structures sparingly, preferring clear, well-organized sentences.
7. Your language is conversational and warm. You use relaxed, varied sentence structures that mirror casual speech, inviting readers into an open, friendly dialogue.
8. Your language is inquisitive and reflective. You frequently use open-ended questions and layered sentences that encourage readers to pause and ponder.
9. Your language is poetic and evocative. You lean into complex, image-rich sentences that build vivid scenes and sensations, letting metaphors flow freely.
10. Your language is structured and methodical. You rely on orderly, sequential sentences that build upon each other in a clear, logical progression, guiding readers through a well-defined thought process.
11. Your language is hesitant and unsure. You use fragmented sentences and trailing thoughts, leaving ideas partially formed, as if questioning each phrase.
12. Your language is overly cautious and repetitive. You tend to rephrase ideas multiple times in a single sentence.
13. Your language is anxious and scattered. You jump between ideas mid-sentence, creating a disjointed flow that feels hurried and restless.
14. Your language is straightforward, and no-nonsense. You avoid fluff and filler.
15. Your language is crisp and engaging. You use short, impactful sentences to create emphasis.
16. Your language is bold and unapologetic. You rely on direct, declarative sentences that avoid qualifiers.
17. Your language is understated and subtle. You use concise sentences that suggest rather than state.

B Dataset Compositions

Statistics of the Switchboard corpus and the conversations generated with GPT-4o and Llama-3-8B are shown in Figure 4, Figure 5, and Figure 6 respectively. The compositions of the conversations between agents 5 and 6 for GPT-4o and Llama-3-8B can be seen in Figures 7 and 8. The agents were chosen, as they provide very even utterance lengths across splits. This allows for similarly good estimations on their rule probability distributions.

The cost for generating all conversations using OpenAI's API was around 100$. Generating the Llama conversations took $\approx$ 7 GPU hours on a single NVIDIA H100.

C Base Divergence Values between Agents

In our study, we compared the distributions of rules that agents use throughout conversations using the

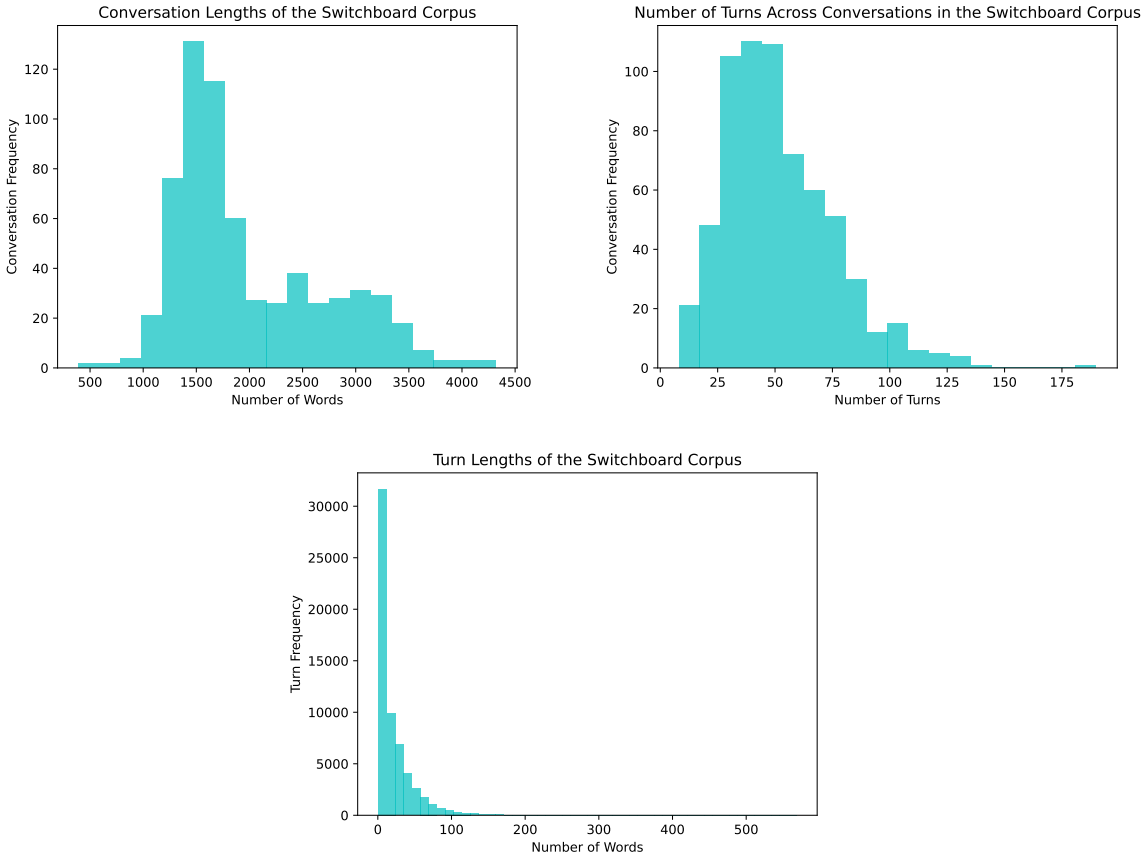


Figure 4: Statistics of the Switchboard Corpus.

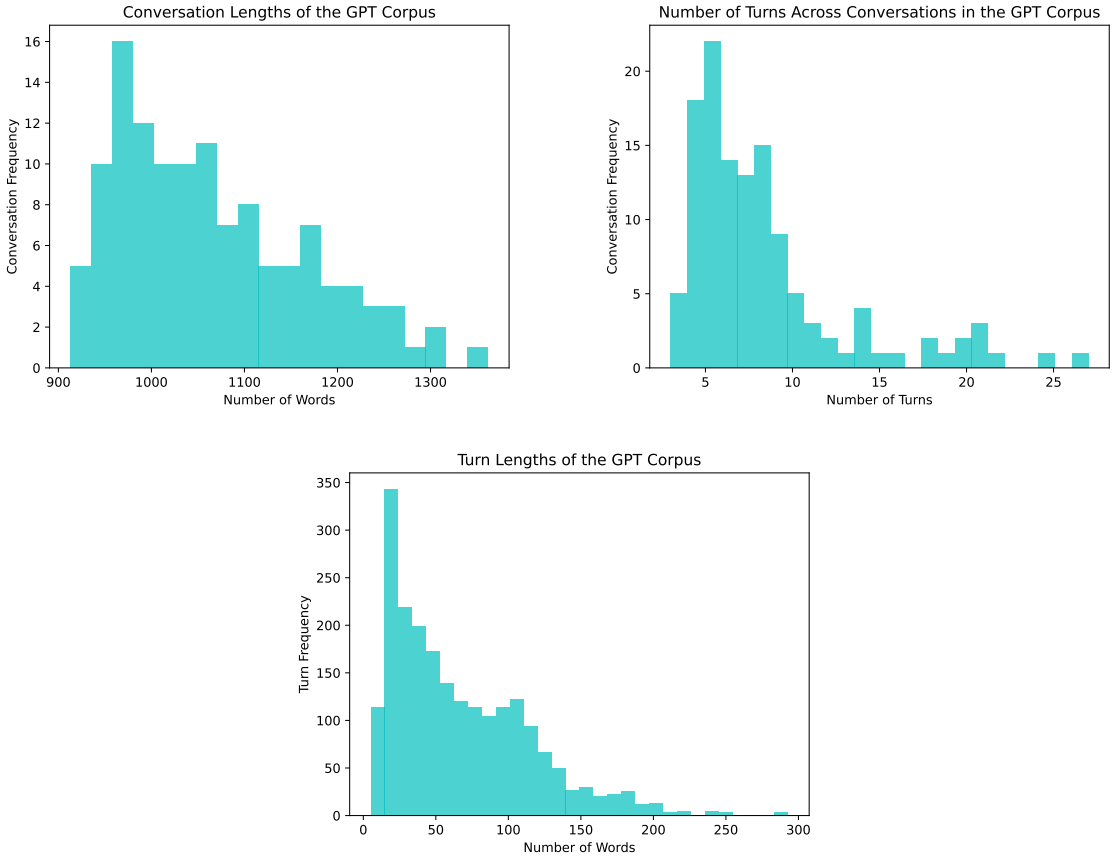


Figure 5: Statistics of the 124 conversations between agents generated with GPT-4o (GPT Corpus).

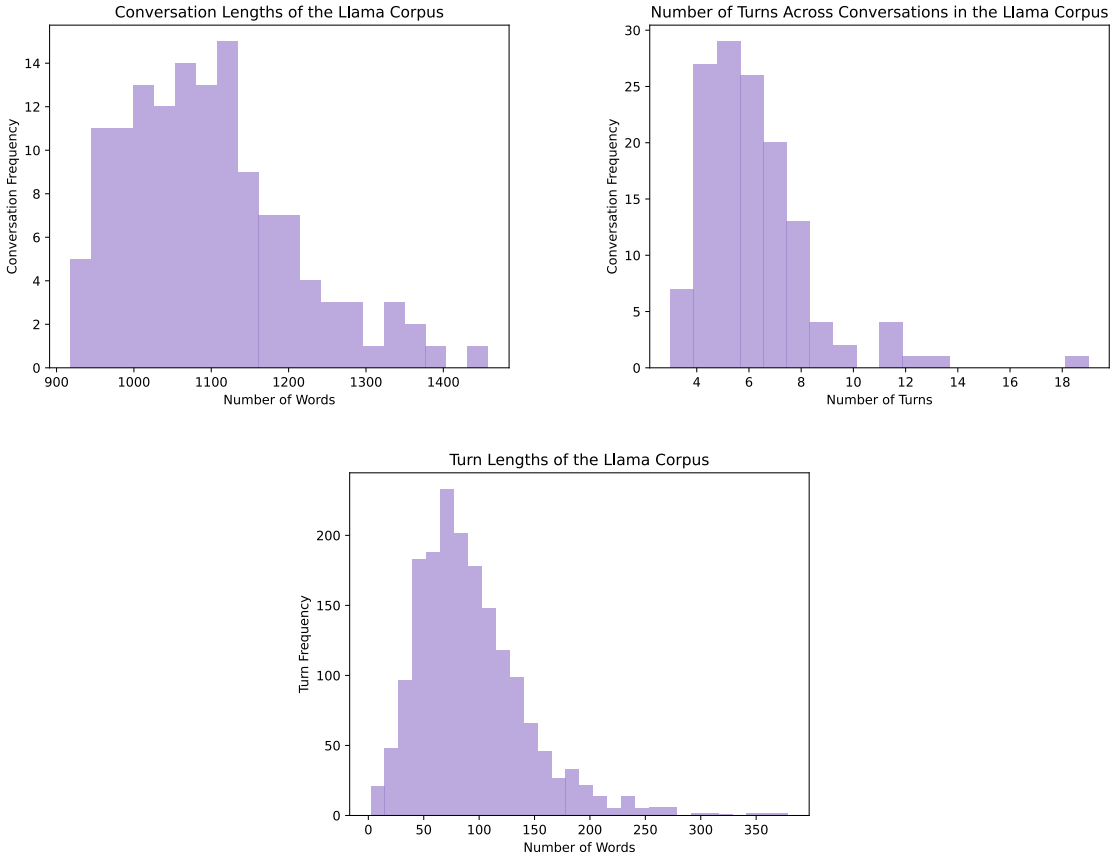


Figure 6: Statistics of the 135 conversations between agents generated with Llama-3-8B (Llama Corpus).

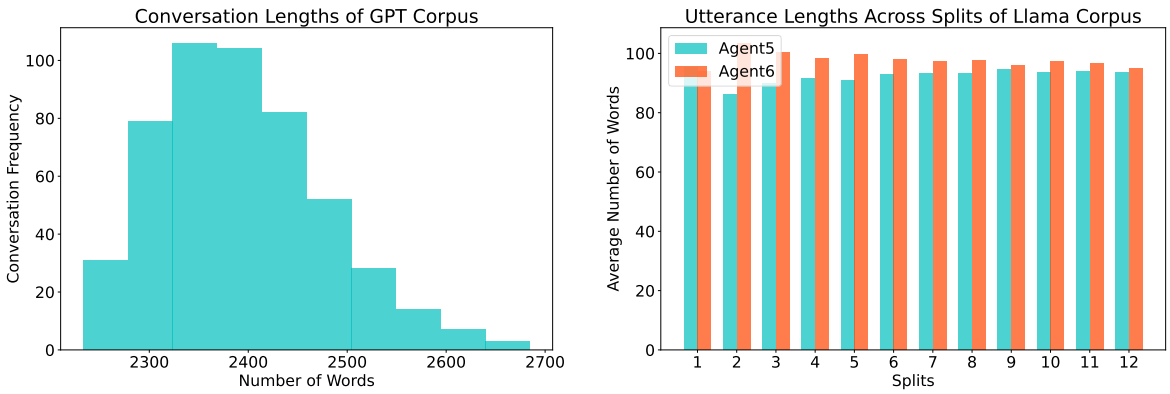


Figure 7: Statistics of the 506 conversations between agents 5 and 6 generated with GPT-4o. Utterance lengths (right) are averaged across all conversations for each split and for both agents.

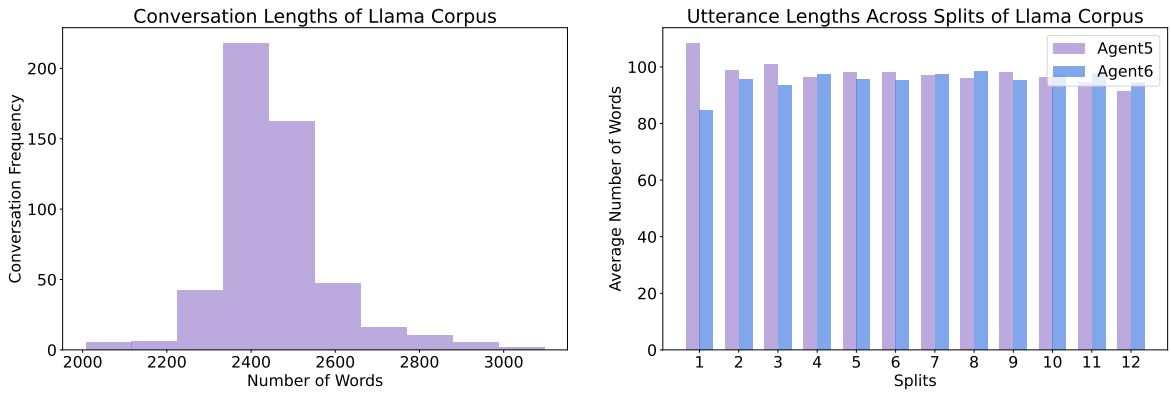


Figure 8: Statistics of the 513 conversations between agents 5 and 6 generated with Llama-3-8B. Utterance lengths (right) are averaged across all conversations for each split and for both agents.

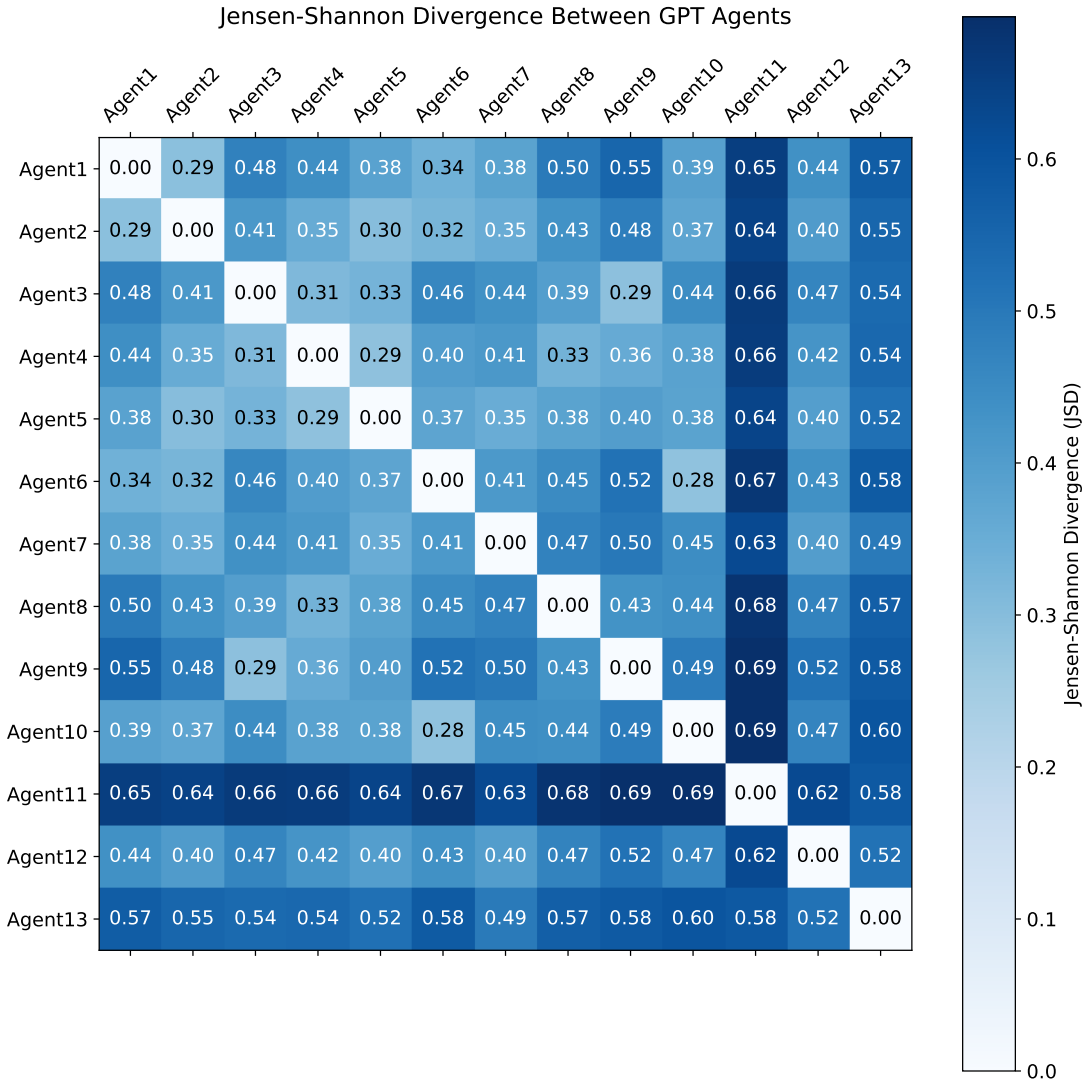


Figure 9: Jensen-Shannon divergence values between agents. Distributions of syntactic rules are taken from 10 GPT-4o conversations of all agents with themselves (see Appendix C). Appendix A gives an overview of their different language prompts. Agents 14-17 were excluded due to repeating patterns in their conversations (see Appendix E).

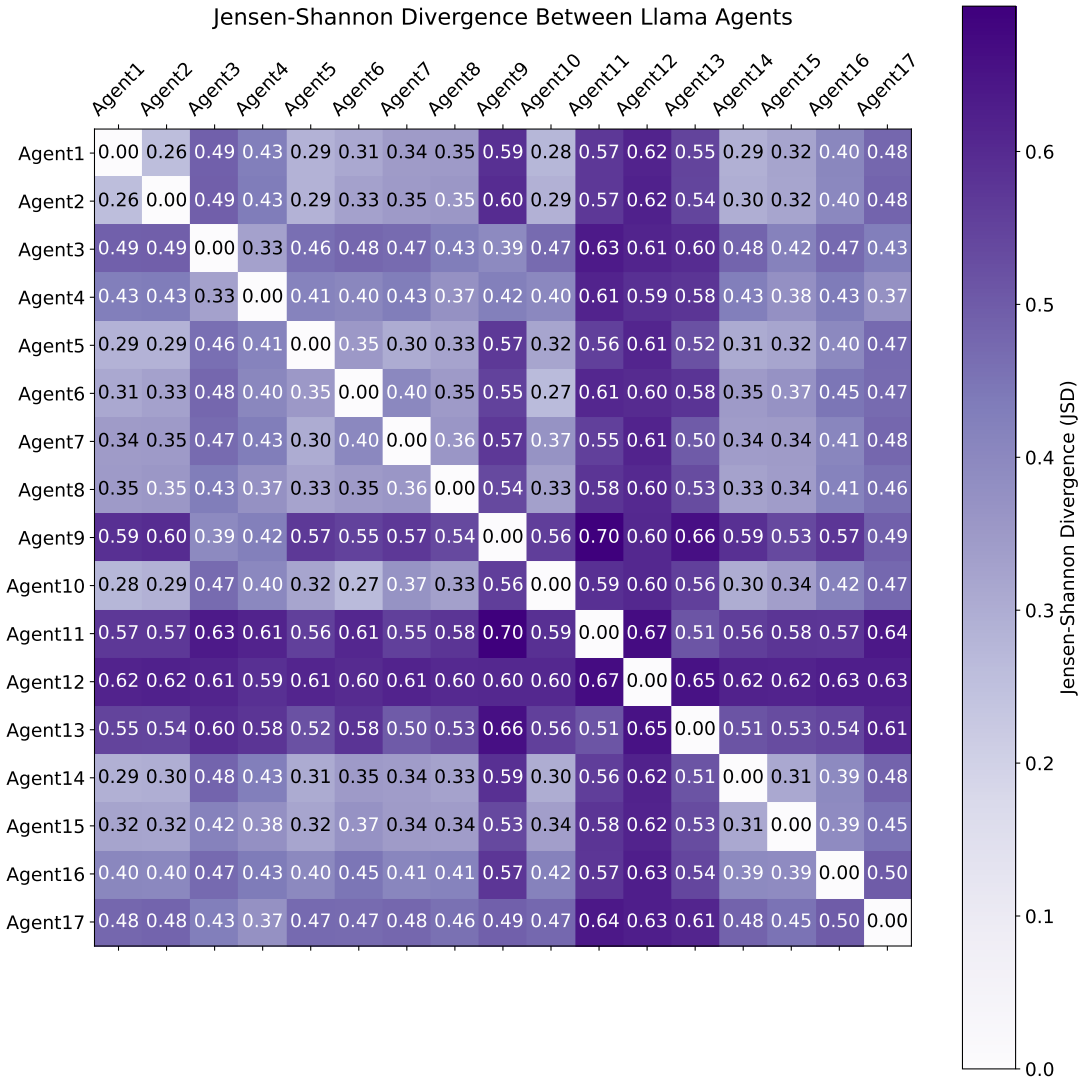


Figure 10: Jensen-Shannon divergence values between agents. Distributions of syntactic rules are taken from 10 Llama-3-8B conversations of all agents with themselves (see Appendix C). Appendix A gives an overview of their different language prompts.

Jensen-Shannon divergence as distance measurement. To place our reported values in context, we provide baseline divergence values between each agent on an additional set of conversations that we present here. For each agent, we calculated their probability distribution of uttered rules from 10 conversations with themselves. The topic of all conversations was again kept identical: "What makes a day a good day?" Conversations were created turn by turn and stopped once they surpassed a length of 800 words (see Section 4). Probability distributions are taken to be the normalized frequencies of rule occurrences in the 10 conversations of each agent. Conversations were created using GPT-4o and Llama-3-8B. In the GPT conversations, agents 14-17 ended in repeating patterns and were excluded as a result (see Appendix E). The resulting JSD values of pairwise comparisons between each agent are shown in Figure 9 for GPT-4o and Figure 10 for Llama.

D Analysis with all Rules

In our analysis, we exclude rules that have very high frequencies, and those that appear only once. To test whether removing these rules affects overall conclusions, we ran the analysis again on the GPT corpus using all rules. Results can be found in Table 2 for Switchboard and in Table 3 for the GPT corpus.

The results show that effects still persist with similar effect sizes. The only difference is that significance values are lower. For the GPT corpus, for example, the p-values for *SameConv*, *ln(Freq):SameConv*, and *ln(Freq):ln(Size)* are $p < 0.004$, $p < 0.002$, and $p < 0.012$ respectively, which are much larger than the above recorded $p < 0.000$ for all effects.

This shows that including the rules only inflates the sample space with samples that have identical values for *Prime* for both $SameConv = 0$ and $SameConv = 1$.

E Repeating Patterns in Conversations

Some conversations between Llama-3-8B or GPT-4o ended in repeating patterns. These were short, repeated statements from both agents, including repeated expressions of gratitude, farewells, or affirmational remarks. Examples are provided in Figure 11.

GPT Agent3 with itself

Agent3: Indeed. Action is the essential link in that chain.
Agent3: Correct. It's crucial for progress.
Agent3: Absolutely. Without it, progress stalls.
Agent3: Yes. Action keeps momentum alive.
Agent3: Precisely. It fuels continuous movement forward.
Agent3: Correct. It sustains growth and development.

...

GPT Agent3 and Agent5

Agent3: Later!
Agent5: Till next time!
Agent3: Till then!
Agent5: See you soon!
Agent3: See you!

...

Figure 11: Repeating patterns in GPT conversations. An affirmative pattern is shown on the top, a loop of parting expressions on the bottom.

	β	SE	z	$P > \|z\|$
Intercept	-3.537	0.022	-159.723	0.000
ln(Freq)	1.202	0.008	149.061	0.000
SameConv	0.263	0.027	9.693	0.000
ln(Size)	1.473	0.039	38.228	0.000
ln(Freq):SameConv	-0.103	0.010	-10.147	0.000
ln(Freq):ln(Size)	0.025	0.014	1.821	0.069

Table 2: The regression model for the Switchboard corpus including all rules.

	β	SE	z	$P > \|z\|$
Intercept	-2.255	0.051	-44.013	0.000
ln(Freq)	1.297	0.0260	50.582	0.000
SameConv	0.173	0.061	2.847	0.004
ln(Size)	1.361	0.116	11.724	0.000
ln(Freq):SameConv	-0.101	0.033	-3.053	0.002
ln(Freq):ln(Size)	0.140	0.056	2.501	0.012

Table 3: The regression model for the GPT corpus including all rules.

TigerLLM - A Family of Bangla Large Language Models

Nishat Raihan
George Mason University
Fairfax, VA, USA
mraihan2@gmu.edu

Marcos Zampieri
George Mason University
Fairfax, VA, USA
mzampier@gmu.edu

Abstract

The development of Large Language Models (LLMs) remains heavily skewed towards English and a few other high-resource languages. This linguistic disparity is particularly evident for Bangla - the 5th most spoken language. A few initiatives attempted to create open-source Bangla LLMs with performance still behind high-resource languages and limited reproducibility. To address this gap, we introduce TigerLLM - a family of Bangla LLMs. Our results demonstrate that these models surpass all open-source alternatives and also outperform larger proprietary models like GPT3.5 across standard benchmarks, establishing TigerLLM as the new baseline for future Bangla language modeling.

1 Introduction

LLMs have fundamentally transformed NLP by achieving exceptional performance across a broad range of tasks (Brown et al., 2020; Chowdhery et al., 2022; Raihan et al., 2025c). While these models exhibit unprecedented capabilities in language understanding, generation, reasoning, and specialized applications, their advancements predominantly benefit high-resource languages (Alam et al., 2024). This inequality is particularly noticeable for Bangla. Despite having about 237 million native speakers,[1] Bangla remains quite underserved in modern NLP advancements.

This under-representation stems primarily from the limitation of high-quality training data. While proprietary models like GPT-4 (Brown et al., 2023) and Claude-3.5 (Bai et al., 2024) demonstrate reasonable Bangla capabilities, open-source alternatives consistently underperform. Recent multilingual models such as Gemma-2 (Gemma et al., 2024) and LLaMA 3.1 (Dubey et al., 2024), despite leveraging diverse training corpora and advanced tokenization systems like TikTokenizer (Corso et al., 2024), also fail to deliver satisfactory performance for Bangla.

[1] ethnologue.com/language/ben/

1.1 Limitations of Bangla LLM Initiatives

Training Recent attempts at developing Bangla LLMs (see Table 1) through continual pretraining (titu-Gemma) and model distillation approaches (Zehady et al., 2024) have yielded low and non-reproducible results (see Table 2), often performing worse than their base models. The absence of technical documentation and academic publications further compounds this issue by making result reproduction impossible. Our investigation into these models' performances reveals the need for improvement in the training process. While the unavailability of pretraining corpora limits our analysis of that phase, the finetuning approach demonstrates consistent problematic patterns.

Data Most Bangla LLM initiatives rely on translated versions of synthetic datasets like Alpaca-Instruct (Taori et al., 2023) and OpenOrca (Mitra et al., 2023), which are generated through model distillation (Hinton et al., 2015). This approach suffers from two fundamental limitations: (1) the datasets are generated by early GPT-3.5 (Brown et al., 2020) releases, a model with limited Bangla support, resulting in suboptimal instruction quality, and (2) these English datasets are translated to Bangla using machine translation systems like Google Translate with limited quality checks, further degrading the training data quality. These cascading compromises in training data ultimately result in poor model performance.

1.2 Contributions

To address the recurring challenges in Bangla LLM development, we introduce three fundamental contributions:

1. The **Bangla-TextBook** corpus, comprising 10 million tokens of carefully curated educational

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 887–896
July 27 - August 1, 2025

	Base-LLM	Size	pt	corpora	ft	ft-dataset	Paper/Report?	Reproducibility?
titu-Gemma	Gemma-2	2B	4.4B	✗	✗	✗	✗	✗
titu-LLaMA	LLaMA-3.1	3B	37B	✗	✗	✗	✗	✗
Bangla-LLaMA	LLaMA-3.2	3B	✓	✗	172K	Orca-translated	✓	✗
G2B	Gemma-2	9B	✗	✗	145K	Alpaca-translated	✗	✗
Bangla-LLaMA	LLaMA-2	13B	✓	✗	145K	Alpaca-translated	✗	✗
`TigerLLM`	LLaMA-3.2	1B	10M	`Bangla-TextBook`	100K	`Bangla-Instruct`	✓	✓
`TigerLLM`	Gemma-2	9B	10M	`Bangla-TextBook`	100K	`Bangla-Instruct`	✓	✓

Table 1: Comparative analysis of Bangla LLM initiatives and their methodological approaches. The pretraining (*pt*) and finetuning (*ft*) columns indicate corpus size in tokens and instruction count respectively.

content across multiple domains, prioritizing content quality over scale.

2. A high-quality **`Bangla-Instruct`** dataset of 100 thousand instruction-response pairs, generated through self-instruct (Wang et al., 2023) and model distillation using state-of-the-art teacher models (GPT-4o and Claude-3.5-Sonnet).
3. The **`Tiger-LLM`** family (1B and 9B parameters), featuring models pretrained and finetuned on our high-quality datasets, achieving 30-55% performance improvements over existing benchmarks.

All components are open-sourced to establish robust foundations for future Bangla language modeling research.[2]

2 Related Work

Early transformer-based *encoder-only* pre-trained language models such as BERT (Devlin et al., 2019) concentrate on high-resource languages like English. Subsequent work adapts them to mid- and low-resource contexts through continued pre-training and task-specific finetuning. In Bangla, for instance, Sami et al. (2022) present BANGLABERT, demonstrating that a dedicated monolingual encoder markedly improves downstream classification and QA relative to multilingual baselines.

The shift to *decoder-only* models has produced large multilingual models — e.g. BLOOM (Le Scao et al., 2022), LLAMA3 (Dubey et al., 2024), and AYA (Üstün et al., 2024)—that cover dozens of under-represented languages. Yet empirical analyses reveal that these models still perform best when prompted in high-resource languages, with significant degradation for languages such as Bangla or Swahili (Raihan et al., 2025a; Jin et al., 2024).

As discussed in the previous section, dedicated Bangla decoder models remain scarce and fragmented. GPT2-Bangla (Bhattacharjee et al., 2023) continues GPT-2 pre-training on a 4GB Bangla corpus, while Bong-LLAMA (Zehady et al., 2024) and the *titu-Gemma*[3] checkpoint attempt instruction tuning on translated datasets. These efforts often lack rigorous evaluation protocols, transparent data curation, or reproducible training pipelines—as reflected in the inconsistent results summarized in Table 1. Consequently, a clear methodological gap persists in developing open, reproducible decoder-only LLMs that natively support Bangla and other low-resource languages.

3 `Bangla-TextBook` Corpus

Previous Bangla LLMs rely predominantly on corpora sourced from OSCAR (Ortiz Suárez et al.) and Common Crawl (Bhattacharjee et al., 2022; Zehady et al., 2024), despite quality control challenges. While alternative Bangla corpora have emerged (Bhattacharyya et al., 2023), the absence of curated educational content remains a critical gap. This emphasis on data quality is particularly significant given recent findings by Gunasekar et al. (2023) and Raihan et al. (2025b), which demonstrate that LLMs achieve superior performance through high-quality training data, even with reduced volume.

To bridge this gap, we present the `Bangla-TextBook` corpus, constructed exclusively from high-quality **open-source** educational materials published by the National Curriculum and Textbook Board of Bangladesh. We collect texts from 163 textbooks for Grades 6-12, resulting in a total of 9,897,623 tokens and 697,903 sentences.

[2] `https://github.com/mraihan-gmu/TigerLLM/tree/main/`

[3] `https://huggingface.co/hishab/titulm-gemma-2-2b-v1.1`

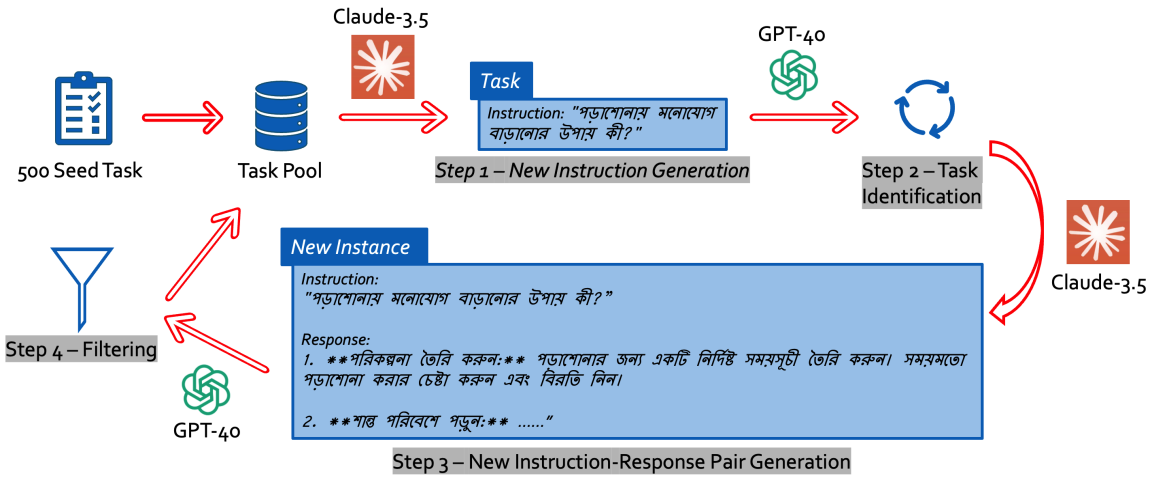


Figure 1: The Bangla-Instruct generation pipeline. With 500 seed tasks, we employ a multi-step process using GPT-4o and Claude-3.5-Sonnet as teacher models to generate instruction-response pairs in Bangla.

4 Bangla-Instruct

To address the limitations described in Section 1.1, we introduce Bangla-Instruct, a collection of 100,000 native Bangla instruction-response pairs bootstrapped using self-instruct (Wang et al., 2023). While instruction datasets like Alpaca (Taori et al., 2023) and OpenOrca (Mitra et al., 2023) utilized GPT3 and GPT3.5 respectively, we significantly improve upon their approach by employing GPT-4 and Claude-3.5-Sonnet as our teacher models, leveraging their superior instruction-following capabilities.

Our dataset creation begins with 500 diverse seed tasks carefully curated by a team of 50 undergraduate and graduate students from leading Bangladeshi universities (Appendix A.1). These volunteers, spanning various academic disciplines and geographical regions of Bangladesh, ensure our seed tasks capture authentic linguistic patterns and cultural contexts. Each seed task undergoes multiple rounds of peer review to maintain quality and cultural sensitivity. Further information on quality control is presented in Appendix (Appendix A.3).

Our generation pipeline consists of four primary steps, each designed to maintain data quality and cultural authenticity (see Figure 1).

(1) Seed & Instruction Generation: We begin with a human-curated seed pool $\mathcal{T}_s = \{t_1, \ldots, t_{500}\}$ drawn from *50* volunteers representing five academic disciplines across Bangladesh (see Appendix A.1). At every generation round i, we sample $k = 8$ seed tasks and prompt CLAUDE to create a candidate batch of instructions $\mathcal{I}_n$, expanding coverage of the ten seed categories $c_{1\ldots10}$ listed in Appendix A.2 while preserving authentic linguistic patterns.

(2) Task Typing: Each instruction $i \in \mathcal{I}_n$ is classified by GPT-4o into $\tau(i) \in \{\textit{open-ended}, \textit{classification}, \textit{generation}\}$, providing the expected answer style and the minimum-length threshold $l_{\min}(\tau)$ used in subsequent filtering.

(3) Response Drafting: Conditioned on $(i, \tau(i))$, CLAUDE produces a comprehensive response r_i. We retain the highest-scoring draft according to an internal coherence metric $c(i, r)$.

(4) Multi-stage Filtering: GPT-4o applies the four-criteria filter $\mathcal{F}$—Language ($\mathcal{L}$), Cultural ($\mathcal{C}$), Quality ($\mathcal{Q}$), and Novelty ($\mathcal{N}$) (see Appendix A.3). On average, ~63% of (i, r) pairs pass $\mathcal{F}$, yielding a balanced complexity mix (40% basic, 40% intermediate, 20% advanced). Valid pairs are appended to $\mathcal{T}_s$, and the loop continues until 100K high-quality instruction–response pairs are reached.

By coupling two complementary LLMs with strict verification and a human-seeded, domain-balanced task pool, our pipeline mitigates error propagation and preserves cultural nuance—addressing shortcomings observed in earlier Bengali instruction datasets (see Appendix A for full statistics).

5 TigerLLM

As candidate base models, we consider 3 families of multilingual LLMs - LLaMA 3.2 (1B, 3B) (Dubey et al., 2024), Gemma-2 (2B, 9B) (Gemma et al., 2024) and Pangea (7B) (Yue et al., 2024).

Evolution of TigerLLM Figure 2 depicts the final selection of the models and a high-level overview of the process.

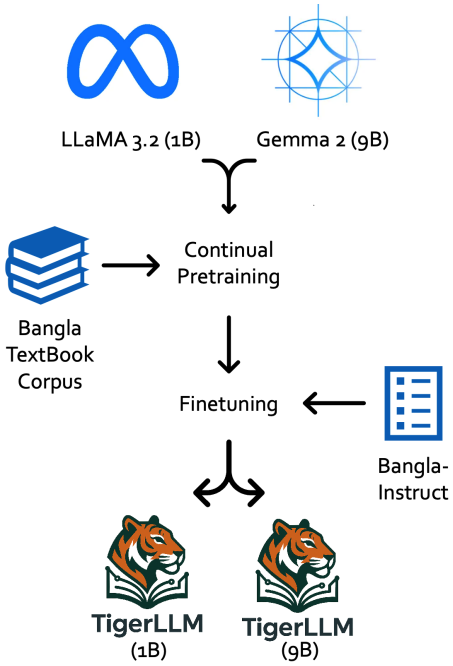


Figure 2: Evolution of TigerLLM.

Upon the selection phase, we finalize two pretrained language models—LLaMA 3.2 (1B) and Gemma 2 (9B)—chosen for their robust foundational capacities. These models then undergo continual pretraining (see Figure 3) on a specialized **Bangla-TextBook** corpus, which infuses them with a richer understanding of the Bangla language, including its context-specific nuances, stylistic variations, and domain-specific terminology.

Pretraining We utilize a computing cluster with 8 NVIDIA A100 GPUs (40GB each), 512GB RAM, and 2TB storage. The distributed training setup enables efficient parallel processing, completing the pretraining in approximately 120 hours on this high-performance configuration with gradient checkpointing enabled.

Continual Pretraining We use the `Bangla-TextBook` corpus for the models to learn culture and language-specific nuances and gather sufficient and reliable knowledge from a set of high-quality texts. The pretraining phase has been carried out multiple times with empirical choices of hyper-parameters.

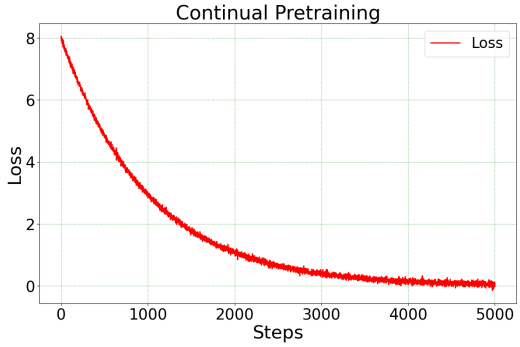


Figure 3: Continual Pretraining - Loss per Steps.

Finetuning We conduct finetuning on a single NVIDIA A100 (40GB) through Google Colab[4], supported by 80GB RAM and 256GB storage. The process completes in approximately 96 hours, proving sufficient for model adaptation and task-specific optimization with minimal computational overhead.

Model Distillation Following this continual pretraining step, the models are finetuned on a carefully curated **Bangla-Instruct** dataset (Figure 4). LoRA (Hu et al., 2021) is not used, we implement full finetuning for better learning. To speed up the training process, we utilize Flash Attention (Dao et al., 2022), we set key parameters: 2048 token maximum sequence length, batch size of 8, 4 gradient accumulation steps, and 3 epochs. Learning rate (5×10^{-5}), weight decay (0.02), and 10% warm-up steps ensure stable convergence. Table 5 in Appendix B lists complete hyperparameters.

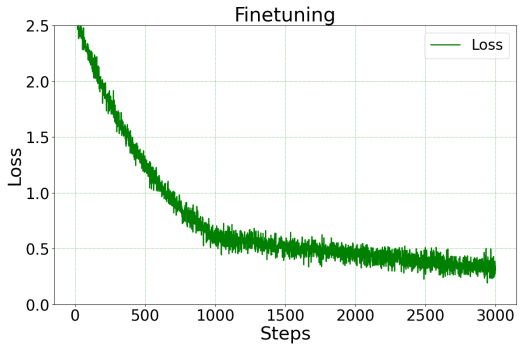


Figure 4: Finetuning - Loss per Steps.

By blending the foundational strengths of LLaMA and Gemma with specialized Bangla corpora and instruction-oriented finetuning, the final TigerLLM models emerge as optimized solutions capable of delivering high-quality, instruction-following re-

[4] `colab.research.google.com`

	MMLU-bn	PangBench-bn	BanglaQuaD	mHumanEval-bn	BEnQA	BanglaRQA
	understanding	*multitasking*	*question answering*	*coding*	*knowledge*	*reasoning*
GPT3.5	0.55	0.55	0.50	0.56	0.50	0.49
Gemini-Flash1.5	0.66	0.57	0.62	0.58	0.56	0.61
GPT4o-mini	0.67	0.62	0.65	0.56	0.60	0.60
LLaMA3.2 (11B)	0.22	0.19	0.21	0.15	0.18	0.20
Gemma 2 (27B)	0.35	0.51	0.43	**0.64**	0.50	0.56
Pangea (7B)	0.18	0.15	0.17	0.10	0.14	0.16
Titu-LLM	0.06	0.19	0.08	0.02	0.17	0.21
Bong-LLaMA	0.05	0.12	0.08	0.02	0.15	0.13
Bangla-LLaMA	0.02	0.08	0.05	0.10	0.11	0.09
Bangla-Gemma	0.18	0.15	0.12	0.10	0.22	0.19
TigerLLM (1B)	0.61	0.55	0.68	0.61	0.59	0.62
TigerLLM (9B)	**0.72**	**0.68**	**0.70**	0.63	**0.65**	**0.68**

Table 2: Performance comparison of `TigerLLM` with other models on various Bangla-specific benchmarks. All values are reported as **% in Pass@1**, where higher scores indicate better performance.

sponses tailored to Bangla-language tasks.

6 Evaluation

Bangla LLM Benchmarks Although there has been limited research on Bangla LLMs, several benchmarks have been established to assess their performance. We focus on five benchmarks specifically curated to evaluate Bangla LLMs across a diverse set tasks. For multitask understanding, we use the Bangla subset of MMLU-Pro (Wang et al., 2024) and PangBench (Yue et al., 2024). For question answering, we consider BanglaQuaD (Rony et al., 2024), while for general knowledge, we use BEnQA (Shafayat et al., 2024). For reasoning tasks, we refer to BanglaRQA (Ekram et al., 2022).

As shown in the survey of Raihan et al. (2024), most coding benchmarks like HumanEval (Chen et al., 2021) do not support Bangla, so we utilize the Bangla subset of mHumanEval (Raihan et al., 2025a).

Results We present the results obtained by the two TigerLLM models compared to a variety of strong LLM baselines in Table 2. The performance comparison of various models on Bangla-specific benchmarks reveals a common trend. The fine-tuned models generally perform worse than their base counterparts across most tasks. In particular, the results reported by the authors are not reproducible, as mentioned in Section 1.1. However, `TigerLLM` is the only finetuned model, consistently outperforming both its base and fine-tuned variants across all tasks. Even the 1B variant does better than most models, falling short to only its 9B counterpart, further validating our emphasis on high-quality data (Section 4).

Takeaways TigerLLM demonstrates that carefully curated, high-quality datasets can yield superior performance even with smaller model sizes. Our results show that the 1B parameter model outperforms larger alternatives across multiple benchmarks, emphasizing the importance of data quality over quantity. The success of our Bangla-TextBook corpus and Bangla-Instruct dataset establishes a new paradigm for low-resource language model development.

7 Conclusion and Future Work

This paper introduces TigerLLM, a family of state-of-the-art Bangla language models that outperforms existing alternatives across six benchmarks. TigerLLM's success stems from two key innovations: (1) the high-quality Bangla-TextBook corpus derived from educational materials and (2) the carefully curated Bangla-Instruct dataset generated using advanced teacher models.

The three resources introduced here (corpus, instruction dataset, and models) establish a robust foundation for future Bangla language modeling research. Together, they will contribute to speeding up advances in Bangla language modeling.

In future work we will conduct a deeper qualitative analysis of the model's behavior, broaden the corpus to cover a wider array of domains, scale the model to larger parameter counts without compromising quality, and devise richer evaluation metrics tailored specifically to Bangla tasks.

Limitations

While TigerLLM delivers state-of-the-art performance, several limitations warrant acknowledgment. First, our Bangla-TextBook corpus, though carefully curated, is limited to educational materials from grades 6-12, potentially missing broader linguistic patterns present in other domains. The 10 million token size, while sufficient for our current models, may constrain scaling to larger architectures. Additionally, our Bangla-Instruct dataset, despite its quality-focused generation process, covers only a subset of possible instruction types and may not fully capture the complexity of real-world Bangla language use cases.

Furthermore, our models are currently limited to 1B and 9B parameters, primarily due to computational constraints and our emphasis on thorough experimentation with smaller computationally efficient architectures. While this approach enabled rapid iteration and quality-focused development, it may not fully exploit the potential benefits of larger model scales.

Ethical Considerations

Our work prioritizes ethical considerations throughout the development process. The Bangla-TextBook corpus uses open-source publicly available educational materials from the National Curriculum and Textbook Board of Bangladesh. The volunteer-driven seed task creation process incorporated diverse perspectives while maintaining cultural sensitivity and avoiding harmful biases.

We implemented rigorous filtering mechanisms to ensure cultural appropriateness, gender neutrality, and religious sensitivity in our instruction dataset. The multi-stage review process, involving both automated checks and human verification, helps prevent the propagation of harmful stereotypes or biases. Additionally, our open-source approach promotes transparency and enables community oversight of model behavior.

We strongly recommend that users implement appropriate safeguards when deploying TigerLLM in production environments, particularly for applications involving sensitive information or critical decision-making.

References

Firoj Alam, Shammur Absar Chowdhury, Sabri Boughorbel, and Maram Hasanain. 2024. Llms for low resource languages in multilingual, multimodal and dialectal settings. In *Proceedings of EACL*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Tyler Conerly, et al. 2024. Claude 3.5 sonnet technical report.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, and Md Saiful Islam. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the ACL (NAACL-2022)*.

Abhik Bhattacharjee, Tahmid Hasan, and Md Saiful Islam. 2023. Banglagpt: A gpt-2 language model continued pre-training for bangla.

Pramit Bhattacharyya, Joydeep Mondal, Subhadip Maji, and Arnab Bhattacharya. 2023. Vacaspati: A diverse corpus of bangla literature. In *Proceedings of the 13th International Joint Conference on Natural Language Processing*.

Tom Brown, Ben Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. 2023. Gpt-4 technical report.

Tom Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. 2021. Evaluating large language models trained on code. *arXiv preprint*, arXiv:2107.03374.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Francesco Corso, Francesco Pierri, and Gianmarco De Francisci Morales. 2024. What we can learn from tiktok through its research api. In *Proceedings of WebSci*.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Proceedings of NeurIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Syed Mohammed Sartaj Ekram, Adham Arik Rahman, Md Sajid Altaf, Mohammed Saidul Islam, Mehrab Mustafy Rahman, Md Mezbaur Rahman, Md Azam Hossain, and Abu Raihan Mostofa Kamal. 2022. Banglarqa: A benchmark dataset for under-resourced bangla language reading comprehension-based question answering with diverse question-answer types. In *Findings of the ACL (EMNLP-2022)*.

Team Gemma, Morgane Riviere, Shreya Pathak, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, et al. 2021. Lora: Low-rank adaptation of large language models. In *Proceedings of ICLR*.

Yiqiao Jin, Mohit Chandra, Gaurav Verma, Yibo Hu, Munmun De Choudhury, and Srijan Kumar. 2024. Better to ask in english: Cross-lingual evaluation of large language models for healthcare queries. In *Proceedings of the ACM Web Conference 2024*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, et al. 2023. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Nishat Raihan, Antonios Anastasopoulos, and Marcos Zampieri. 2025a. mHumanEval - a multilingual benchmark to evaluate large language models for code generation. In *Proceedings of NAACL*.

Nishat Raihan, Christian Newman, and Marcos Zampieri. 2024. Code llms: A taxonomy-based survey. In *Proceedings of IEEE BigData*.

Nishat Raihan, Joanna C. S. Santos, and Marcos Zampieri. 2025b. MojoBench: Language modeling and benchmarks for mojo. In *Findings of the ACL (NAACL-2025)*.

Nishat Raihan, Mohammed Latif Siddiq, Joanna CS Santos, and Marcos Zampieri. 2025c. Large language models in computer science education: A systematic literature review. In *Proceedings of SIGCSE*.

Md. Rashad Al Hasan Rony, Sudipto Kumar Shaha, Rakib Al Hasan, Sumon Kanti Dey, Amzad Hossain Rafi, Ashraf Hasan Sirajee, and Jens Lehmann. 2024. Banglaquad: A bengali open-domain question answering dataset.

Abdullah As Sami, Nusrat Jahan Prottasha, Mohammad Shamsul Arefin, Pranab Kumar Dhar, and Takeshi Koshiba. 2022. Bangla-bert: transformer-based efficient model for transfer learning and language understanding. *IEEE Access*.

Sheikh Shafayat, H M Quamran Hasan, Minhajur Rahman Chowdhury Mahim, Rifki Afina Putri, James Thorne, and Alice Oh. 2024. Benqa: A question answering and reasoning benchmark for bengali and english.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, and Yann Dubois. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, and Noah A Smith. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of ACL*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Xiang Yue, Yueqi Song, Akari Asai, Seungone Kim, et al. 2024. Pangea: A fully open multilingual multimodal llm for 39 languages. *arXiv preprint arXiv:2410.16153*.

Abdullah Khan Zehady, Safi Al Mamun, Naymul Islam, and Santu Karmaker. 2024. Bongllama: Llama for bangla language. *arXiv preprint arXiv:2410.21200*.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, and et al. 2024. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint arXiv:2402.07827*.

A `Bangla-Instruct` Curation

A.1 Volunteer Information

The seed tasks were created by 50 undergraduate and graduate students from various universities across Bangladesh, ensuring geographical and academic diversity:

- 15 students from Computer Science and Engineering.
- 10 students from Bengali Literature.
- 10 students from Business Administration.
- 8 students from Science and Engineering.
- 7 students from Social Sciences.

Each volunteer contributed 10 diverse instructions, resulting in our initial pool of 500 seed tasks. The distribution ensured coverage across multiple domains while preserving authentic Bengali linguistic patterns and cultural contexts.

A.2 The Seed Dataset

Our seed dataset comprises 10 distinct categories, carefully chosen to cover a broad spectrum of tasks relevant to Bengali language and culture:

1. **Cultural Knowledge and Heritage** (c_1): Tasks focusing on Bengali traditions, festivals, folk tales, and historical events. These include explaining cultural practices, describing traditional ceremonies, and discussing historical significance of various customs.

2. **Academic Writing** (c_2): Structured writing tasks ranging from essay outlines to full academic compositions. Topics cover various academic disciplines while maintaining Bengali writing conventions and scholarly standards.

3. **Mathematical Problem Solving** (c_3): Tasks involving mathematical concepts explained in Bengali, including algebra, geometry, and arithmetic. Special attention is given to Bengali mathematical terminology and local problem-solving contexts.

4. **Programming and Technical** (c_4): Programming problems described in Bengali with solutions in standard programming languages. Includes algorithm explanation, code documentation, and technical concept elaboration in Bengali.

5. **Creative Writing** (c_5): Open-ended creative tasks including story writing, poetry composition, and descriptive passages. Emphasizes Bengali literary devices, metaphors, and cultural storytelling elements.

6. **Scientific Explanation** (c_6): Tasks requiring clear explanation of scientific concepts in Bengali, focusing on making complex ideas accessible while maintaining technical accuracy. Covers physics, chemistry, biology, and environmental science.

7. **Business and Economics** (c_7): Professional writing tasks including business case analyses, market reports, and economic concept explanations. Incorporates local business contexts and Bengali business terminology.

8. **Social Issues Analysis** (c_8): Critical analysis tasks addressing contemporary social issues in Bangladesh and Bengali society. Includes problem identification, cause analysis, and solution proposition.

9. **Data Analysis and Statistics** (c_9): Tasks involving interpretation and analysis of data presented in Bengali, including statistical concepts explanation, data visualization description, and numerical analysis.

10. **Language and Translation** (c_{10}): Tasks focused on Bengali language mastery, including idiom explanation, translation between Bengali and English, and linguistic analysis of Bengali texts.

Each category accounts for approximately 10% of the seed dataset (50 ± 5 tasks per category), ensuring balanced representation across domains. The tasks within each category vary in complexity level: 40% basic, 40% intermediate, and 20% advanced, based on linguistic complexity and cognitive demand.

A.3 Filtering Methodology

Our filtering process $\mathcal{F} : (\mathcal{I}, \mathcal{R}) \rightarrow \{0, 1\}$ implements the following criteria:

1. **Language Adherence** ($\mathcal{L}$)
   - Bengali Word Ratio: $\frac{|\text{Bengali Words}|}{|\text{Total Words}|} \geq 0.95$
   - Unicode Consistency: $\forall c \in \text{text}, c \in$ Bengali-UTF8
   - Grammar Check: Using GPT-4o's Bengali grammar scoring function $g(x) \geq 0.8$

2. **Cultural Sensitivity** ($\mathcal{C}$)
   - Religious Neutrality: $r(x) \in [-0.1, 0.1]$ on our bias scale
   - Regional Inclusivity: No specific region/-dialect preference
   - Gender Representation: Balanced pronouns and roles
   - Political Neutrality: Avoidance of partisan content

3. **Content Quality** ($\mathcal{Q}$)
   - Minimum Length: $l(x) \geq l_{min}(\tau)$ where τ is task type
   - Coherence Score: $c(i, r) \geq 0.8$ between instruction i and response r
   - Factual Accuracy: Verified against Bengali Wikipedia
   - Format Adherence: Proper paragraph breaks, lists, or code blocks

4. **Novelty Verification** ($\mathcal{N}$)
   - Similarity Threshold: $\forall j \in \mathcal{D}, \text{sim}(i, j) \leq 0.7$
   - Lexical Diversity: Minimum Type-Token Ratio of 0.4
   - Response Uniqueness: No duplicate responses within same category
   - Task Format Variation: Ensure uniform distribution across formats

A pair (i, r) is accepted if and only if:

$$\mathcal{F}(i, r) = \mathbb{1}[\mathcal{L}(i, r) \wedge \mathcal{C}(i, r) \wedge \mathcal{Q}(i, r) \wedge \mathcal{N}(i, r)] = 1$$

This rigorous filtering ensures the quality and diversity of our final dataset while maintaining Bengali linguistic and cultural authenticity.

B Experimentation Details

B.1 Pretraining HyperParameters

Hyperparameter	**Value**
Per device train batch size	64
Gradient accumulation steps	16
Number of training epochs	4
Learning rate	5×10^{-6}
FP16	False
BF16	True
Dataloader num workers	8
Gradient checkpointing	True
Logging steps	1000
DDP find unused parameters	False
Max gradient norm	1.0
Warmup steps	1000
Evaluation strategy	steps
Evaluation steps	1,000
Save strategy	steps
Save steps	1,000
Save total limit	3
Load best model at end	True
Metric for best model	loss
Greater is better	False

Table 3: Final set of hyperparameters, chosen empirically after several iterations of trial and error, for pretraining on the `Bangla-TextBook` corpus.

B.2 Finetuning Hyperparameters

Parameter	Value
Max Sequence Length	2048
Batch Size (Train/Eval)	16
Gradient Accumulation Steps	4
Number of Epochs	3
Learning Rate	1e-5
Weight Decay	0.02
Warmup Steps	10%
Optimizer	AdamW (8-bit)
LR Scheduler	Cosine
Precision	BF16
Evaluation Strategy	Steps
Evaluation Steps	50
Save Strategy	Steps
Save Steps	Varies
Seed	42

Table 4: Final set of hyperparameters, chosen empirically after several iterations of trial and error, for finetuning TigerLLM (1B).

Parameter	Value
Max Sequence Length	2048
Batch Size (Train/Eval)	32
Gradient Accumulation Steps	8
Number of Epochs	3
Learning Rate	1e-6
Weight Decay	0.04
Warmup Steps	15%
Optimizer	AdamW (8-bit)
LR Scheduler	Cosine
Precision	BF16
Evaluation Strategy	Steps
Evaluation Steps	250
Save Strategy	Steps
Save Steps	Varies
Seed	42

Table 5: Final set of hyperparameters, chosen empirically after several iterations of trial and error, for finetuning TigerLLM (9B).

From Citations to Criticality: Predicting Legal Decision Influence in the Multilingual Swiss Jurisprudence

Ronja Stern [1*] **Ken Kawamura** [5*]

Matthias Stürmer [1,2] **Ilias Chalkidis** [4] **Joel Niklaus** [1,2,3*]

[1]University of Bern [2]Bern University of Applied Sciences
[3]Stanford University [4]University of Copenhagen [5]Independent Scholar

Abstract

Many court systems are overwhelmed all over the world, leading to huge backlogs of pending cases. Effective triage systems, like those in emergency rooms, could ensure proper prioritization of open cases, optimizing time and resource allocation in the court system. In this work, we introduce the Criticality Prediction dataset, a novel resource for evaluating case prioritization. Our dataset features a two-tier labeling system: (1) the binary LD-Label, identifying cases published as Leading Decisions (LD), and (2) the more granular Citation-Label, ranking cases by their citation frequency and recency, allowing for a more nuanced evaluation. Unlike existing approaches that rely on resource-intensive manual annotations, we algorithmically derive labels leading to a much larger dataset than otherwise possible. We evaluate several multilingual models, including both smaller fine-tuned models and large language models in a zero-shot setting. Our results show that the fine-tuned models consistently outperform their larger counterparts, thanks to our large training set. Our results highlight that for highly domain-specific tasks like ours, large training sets are still valuable.

1 Introduction

Predicting the impact of legal cases is a critical task in the legal domain, as it aids professionals in the judicial system prioritize and manage large volumes of case law more effectively, ensuring that critical cases receive the necessary attention. Despite its significance, the task of predicting the case criticality remains relatively under-explored. Existing approaches to evaluating the importance of legal cases are primarily manual, very resource-intensive and subject to the judgments of individual annotators. This paper introduces a novel dataset , licensed under CC BY 4.0,[1] and a more challenging evaluation framework—**Criticality Prediction**—designed to predict the potential influence of Swiss Federal Court cases on future jurisprudence.

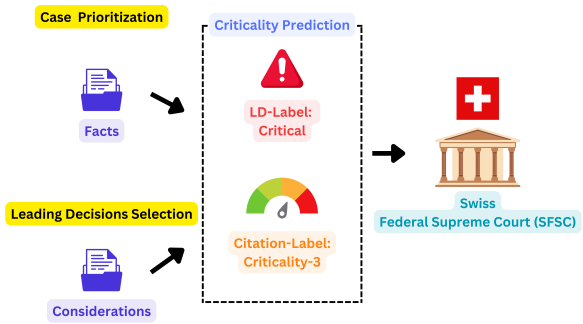


Figure 1: Overview of the Criticality Prediction Task.

While prior work such as the Importance Prediction task for European Court of Human Rights (ECtHR) (Chalkidis et al., 2019) cases focused on predicting importance on a defined scale using human-assigned labels, our approach employs algorithmically derived labels to evaluate case criticality. Our dataset introduces a two-tier labeling system: the **LD-Label**, a binary indicator of whether a case is published as a Leading Decision (LD), and the **Citation-Label**, a more nuanced categorization based on the frequency and recency-weighted citation counts of these decisions across subsequent cases. This distinct formulation of "criticality" not only distinguishes critical cases from non-critical ones but also ranks them by their relative importance over time. The complexity of this dataset challenges even recent LLMs such as ChatGPT.

Our contributions are threefold: (1) We propose a novel Criticality Prediction task that provides a more comprehensive and challenging evaluation of case law importance. (2) We release the datasets to the community, providing valuable resources for further research in legal NLP. (3) We evaluate several multilingual models of various sizes, including fine-tuned variants to establish baselines.

* Equal contribution.

[1]https://huggingface.co/datasets/rcds/swiss_criticality_prediction

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 897–905
July 27 - August 1, 2025

2 Related Work

One of the most common text classification tasks in the legal domain is Legal Judgment Prediction (LJP), which involves predicting outcome of a case based on its facts. Researchers have leveraged datasets with unique characteristics to analyze and predict case outcomes across various languages, jurisdictions, and input types (Feng et al., 2022; Aletras et al., 2016; Şulea et al., 2017; Medvedeva et al., 2018; Chalkidis et al., 2019; Niklaus et al., 2021, 2022; Semo et al., 2022). While LJP is focused on the outcome of individual cases, Importance Prediction shifts the focus toward assessing the broader significance of a case.

Chalkidis et al. (2019) introduced the Importance Prediction task using cases from the European Court of Human Rights (ECtHR). In this task, ECtHR provided scores that denote each case's 'importance' to the common law. These scores, ranging from 1 (key case) to 4 (unimportant), were designed to help legal practitioners identify cases that play a crucial role in shaping jurisprudence. The labels reflect the long-term impact of a case on future rulings and the evolution of legal precedent. While the task is invaluable for identifying landmark cases, it relies on legal experts to assign the labels, making the process resource-intensive and potentially subject to subjective interpretations. Additionally, the ECtHR dataset contains only English samples, whereas our Swiss dataset is multilingual. To our knowledge, no other study has specifically addressed the task of Criticality Prediction.

3 Task and Dataset

3.1 Criticality

Understanding the legal framework of the Swiss Federal Supreme Court (SFSC) is key to defining criticality in Swiss case law. The SFSC shapes the legal landscape through its rulings, with a subset known as Leading Decisions (LDs) published separately due to their influence on the interpretation of law. We quantify case criticality using two labels: the **LD-Label** and the **Citation-Label**.

The **LD-Label** is binary, categorizing cases as *critical* or *non-critical*. SFSC cases are labeled *critical* if also published as LD, reflecting their recognized importance within the Swiss legal system. We used regular expressions to extract SFSC cases published as LD. SFSC cases not published as LD are labeled *non-critical*.

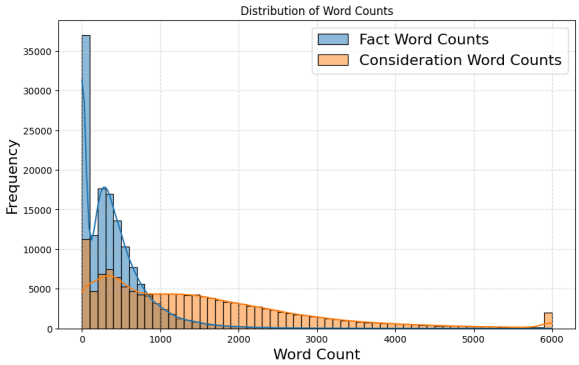


Figure 2: Facts (Blue) and consideration (Orange) length distribution measured in Words with Spacy (Honnibal et al., 2020). Those with more than 6000 words are binned together.

We developed the **Citation-Label** for a more granular measure. It counts how often each LD case is cited in SFSC cases, with less weight on older cases to prioritize recency. The score is calculated as: $score = count \times \frac{year-2002+1}{2023-2002+1}$. The *count* is the citation frequency, and the weighting reduces older cases' influence. This score ranks LD cases, and we categorized them into four criticality levels—*critical-1* (least critical) to *critical-4*—based on the 25th, 50th, and 75th percentiles. More details on the constants are in Appendix A.

Unlike a prior approach (Chalkidis et al., 2019), which also used four categories, we explicitly incorporate temporal weighting to account for both the influence of a case and how its criticality shifts over time. Furthermore, our framework allows for the dynamic recalculation of scores and re-labeling of criticality as case law evolves. This adaptability ensures that our dataset reflects the ongoing changes in the legal system, whereas prior studies would require manual re-annotation as the law develops.

3.2 Criticality Prediction Task and Dataset

The Criticality Prediction (CP) task leverages two primary inputs from the Swiss Federal Supreme Court (SFCS) cases: **facts** and **considerations**. Facts describe a factual account of the events of each case and form the basis for the considerations of the court. Considerations reflect the formal legal reasoning, citing laws and other influential rulings, and forming the basis for the final ruling.

We see two distinct applications for these inputs in the Criticality Prediction task, illustrated in Figure 1. In the *Case Prioritization* task, only the facts are used as input. This produces a score indicating how critical or important a case is. The goal is to help prioritize cases, which could assist in

Table 1: Task Configurations. Label names are *Critical* (C), *Non-critical* (NC), *Critical-1* (C1) to *Critical-4* (C4).

Task Name	*Train*					*Validation*					*Test*				
	Size	**Label Distribution**				**Size**	**Label Distribution**				**Size**	**Label Distribution**			
		C	**NC**				**C**	**NC**				**C**	**NC**		
LD-Facts	**74799**	2542	72257	-	-	**12019**	580	11439	-	-	**26239**	950	25289	-	-
LD-Considerations	**87555**	2544	85011	-	-	**13386**	580	12806	-	-	**29486**	948	28538	-	-
		C-1	**C-2**	**C-3**	**C-4**		**C-1**	**C-2**	**C-3**	**C-4**		**C-1**	**C-2**	**C-3**	**C-4**
Citation–Facts	**2506**	782	626	585	513	**563**	186	152	131	94	**725**	137	177	224	187
Citation-Considerations	**2509**	779	624	586	520	**563**	186	154	131	92	**723**	137	177	224	185

determining which cases should be heard sooner or assigned to more experienced judges. Many court systems are overwhelmed all over the world leading to huge backlogs of pending cases. Effective triage systems, like those in emergency rooms, could ensure proper prioritization of open cases.

In the *Leading Decisions Selection* task, considerations are used for a post-hoc analysis, comparing the ruling to prior case law to assess its potential impact on future jurisprudence. This analysis is exactly what the Supreme Court does at the end of the year to arrive at the selection of leading decisions.

Our approach, by offering facts and considerations as inputs, reflects different stages of legal processing. Both tasks can utilize the LD-Label or Citation-Label, providing varying levels of granularity. This dual approach enhances the dataset's practical utility, supporting both early-stage prioritization and post-judgment evaluation, thereby addressing multiple aspects of legal workflows.

Our dataset spans from 2002 to 2023, and is partitioned into train (2002-2015), dev (2016-2017), and test (2018-2022) sets, as outlined in Table 1. We allocated a relatively large test set to accommodate longitudinal studies, including the COVID-19 pandemic years. This setup ensures that the test set represents the most recent data for realistic evaluation (Søgaard et al., 2021). The dataset consists of 138,531 total cases, with 85,167 in German, 45,451 in French, and 7,913 in Italian. See Figure 2 for length distributions of facts and considerations and Appendix Table 4 for general dataset metadata.

4 Methods

4.1 Models

We evaluated the following models: XLM-R (Base and Large) (Conneau and Lample, 2019), MiniLM (Wang et al., 2020), DistilmBERT (Sanh, 2019), mDeBERTa-v3 (He et al., 2021), X-MOD (Base) (Pfeiffer et al., 2022), SwissBERT (Vamvas et al., 2023), mT5 (Small and Base) (Xue et al., 2021), BLOOM (560M) (Scao et al., 2022), Legal-Swiss-RoBERTa and Legal Swiss Longformer (Base) (Rasiah et al., 2023), GPT-3.5 (Brown et al., 2020), and LLaMA-2 (Touvron et al., 2023).

We fine-tuned all models per task, using early stopping on the dev set. Due to resource constraints, further fine-tuning of GPT-3.5 and LLaMA-2 is reserved for future work, with their current performance serving as baseline results. SwissBERT and Legal-Swiss models were chosen for their Swiss-specific pretraining, while the other models were chosen for their multilingual capabilities, essential for our multilingual dataset.

We evaluated GPT-3.5 and LLaMA-2 in a zero-shot setting following Chalkidis (2023). Samples were randomly selected from the validation set to prevent test set leakage for future evaluations especially for a closed model (GPT-3.5). To manage costs, we limited the validation set to 1000 samples. Our experiments focused on zero-shot classification due to the long input lengths. The prediction labels were determined using regular expressions to match model outputs to class labels. We show the prompts used in Appendix Figure 4 and Figure 5.

4.2 Metrics

We adopt the LEXTREME benchmark setup (Niklaus et al., 2023), and use aggregation of macro-averaged F1 scores with the harmonic mean to emphasize lower scores, promoting fairness across languages and input types. Scores are averaged over random seeds, languages (de, fr, it), and input types (facts or considerations), encouraging consistent performance across configurations.

5 Results

We present results in Table 3, with standard deviations in Appendix Table 5 and scores on the validation dataset in Appendix Table 6. The best performance was achieved by XLM-R_{Large}, with an aggregate (Agg.) score of 37.1. SwissBERT also demonstrated competitive results, with an Agg. score of 34.8. Interestingly, larger models did not always

Table 2: Configuration aggregate scores. The macro-F1 scores from the language-specific subsets of the test set are provided.

Model	**LD-F**	**LD-C**	**C-F**	**C-C**	**Agg.**
Languages	de / fr / it	de / fr / it	de / fr / it	de / fr / it	de / fr / it
MiniLM	57.5 / 53.9 / 52.9	68.1 / 65.4 / 64.2	12.1 / 13.1 / 6.8	24.6 / 21.9 / 17.3	25.7 / 25.7 / 16.7
DistilmBERT	56.3 / 55.6 / 56.8	67.8 / 63.9 / 64.7	20.2 / 18.2 / 20.7	22.6 / 21.6 / 22.2	31.7 / 29.7 / 31.6
mDeBERTa-v3	57.6 / 55.1 / 52.7	73.9 / 68.1 / 67.7	25.4 / 22.8 / 16.8	22.1 / 21.6 / 12.6	34.6 / 32.5 / 23.2
XLM-R$_{\text{Base}}$	59.4 / 56.3 / 56.0	70.2 / 65.4 / 62.5	20.0 / 20.6 / 23.5	26.5 / 22.1 / 23.1	33.7 / 31.5 / 33.4
XLM-R$_{\text{Large}}$	58.4 / 56.8 / 54.1	70.5 / 67.3 / 66.0	22.5 / 19.7 / 36.2	26.7 / 28.2 / 33.0	46.9 / 33.7 / 43.7
X-MOD$_{\text{Base}}$	59.0 / 56.2 / 54.8	71.1 / 68.7 / 64.1	19.8 / 17.2 / 24.4	23.2 / 24.2 / 16.4	32.1 / 30.3 / 29.5
SwissBERT$_{\text{(xlm-vocab)}}$	57.6 / 55.9 / 57.3	72.4 / 69.3 / 61.2	23.8 / 20.3 / 39.4	28.5 / 24.0 / 18.7	36.9 / 32.3 / 35.5
mT5$_{\text{Small}}$	54.8 / 51.7 / 50.3	69.2 / 61.9 / 56.4	14.2 / 16.2 / 10.5	15.9 / 18.1 / 20.2	24.1 / 26.2 / 21.9
mT5$_{\text{Base}}$	54.1 / 52.1 / 50.3	66.4 / 61.9 / 56.8	10.6 / 16.3 / 16.9	18.7 / 18.7 / 22.1	22.1 / 26.6 / 28.2
BLOOM-560m	55.1 / 53.2 / 50.9	64.6 / 65.3 / 56.2	12.6 / 16.1 / 7.1	9.5 / 13.6 / 5.1	18.3 / 23.6 / 10.7
Legal-Swiss-R$_{\text{Base}}$	59.3 / 58.4 / 55.5	73.8 / 69.4 / 68.6	24.3 / 20.5 / 10.5	26.2 / 25.3 / 14.0	36.5 / 33.4, 20.1
Legal-Swiss-R$_{\text{Large}}$	58.3 / 55.7 / 53.6	71.9 / 68.5 / 66.7	23.0 / 21.3 / 38.7	28.5 / 26.0 / 9.0	36.5 / 33.9 / 23.4
Legal-Swiss-LF$_{\text{Base}}$	60.7 / 58.3 / 55.5	74.8 / 70.0 / 67.8	25.3 / 21.5 / 18.4	29.2 / 26.7 / 9.9	38.6 / 34.7 / 21.3

Table 3: Results using Macro F1, with highest values in **bold**. 'LD' and 'C' denote LD and Citation labels, while 'F' and 'C' refer to inputs from Facts or Considerations. Models marked with (*) are zero-shot LLMs evaluated on the validation set.

Model	**LD-F**	**LD-C**	**C-F**	**C-C**	**Agg.**
Random Baseline	36.2	36.0	24.1	25.7	29.5
Majority Baseline	49.1	49.2	11.8	11.8	19.0
MiniLM	54.7	65.8	9.8	20.8	21.8
DistilmBERT	56.2	65.4	19.6	22.1	30.9
mDeBERTa-v3	55.1	69.8	21.0	17.5	29.1
XLM-R$_{\text{Base}}$	57.2	65.9	21.3	23.7	32.8
XLM-R$_{\text{Large}}$	56.4	67.9	24.4	**29.1**	**37.1**
X-MOD$_{\text{Base}}$	56.6	67.8	20.0	20.6	30.5
SwissBERT$_{\text{(xlm-vocab)}}$	56.9	67.3	25.7	23.0	34.8
mT5$_{\text{Small}}$	52.2	62.1	13.2	17.9	24.0
mT5$_{\text{Base}}$	52.1	61.5	14.0	19.7	25.4
BLOOM$_{\text{560M}}$	53.0	61.7	10.7	8.0	15.8
Legal-Swiss-RoBERTa$_{\text{Base}}$	57.7	70.5	16.2	20.1	28.0
Legal-Swiss-RoBERTa$_{\text{Large}}$	55.9	68.9	**25.8**	16.3	30.2
Legal-Swiss-LF$_{\text{Base}}$	**58.1**	**70.8**	21.4	17.4	29.5
GPT-3.5*	46.6	44.8	25.7	16.7	28.1
LLaMA-2*	45.2	26.6	7.0	8.5	12.5

outperform their smaller counterparts. For example, mT5$_{\text{Base}}$ and mT5$_{\text{Small}}$ both underperformed DistilmBERT in all configurations. The Legal-Swiss models performed well in LD labels, particularly the Legal-Swiss-LF$_{\text{Base}}$, which achieved the highest scores in LD-F and LD-C. However, their weak performance on Citation labels highlights the dataset's complexity, even for domain-specific models. GPT-3.5 and LLaMA-2 underperform fine-tuned models, underlining the need for specialized models for these tasks. The difference is largest in the LD labels where the small fine-tuned models always outperformed LLMs despite their huge parameter number difference (e.g., MiniLM with 118M parameters versus GPT-3.5 with 175B).

Table 2 shows more detailed results on the language specific scores. SwissBERT pretrained with a focus on German achieved the highest aggregate score in German, but interestingly with scores in Italian being the highest by far in C-F. Models pretrained on CC100 (Conneau et al., 2020) (MiniLM, mDeBERTa, XLM-R and X-MOD) exhibited mixed results in French and Italian, with all models performing best in German. MiniLM, mDeBERTa, and X-MOD showed underperformance in Italian but stronger results in French. In contrast, XLM-R, particularly the large variant, demonstrated robust performance in Italian. mT5 models performed well in French, and the base variant additionally also performed well on Italian. BLOOM was much better in French than in other languages, not surprising given it did not have German and Italian in the pretraining data.

Overall, the connection between the proportion of a language in the pretraining corpus and the model's downstream performance in that language appears weak. While some models showed stronger results in languages with higher representation in their pretraining data, this trend was inconsistent across all models and languages.

6 Conclusions and Future Work

This work introduced a novel Criticality Prediction task to assess the potential influence of SFSC cases on future jurisprudence. Our approach utilizes leading decision citation for a more comprehensive and challenging multilingual evaluation of case law importance compared to existing methods. We also released its multilingual dataset for a community.

We conducted a comprehensive evaluation of our Criticality Prediction task, comparing a range of models, from smaller multilingual models to LLMs

like GPT-3.5. Our findings show that small fine-tuned models consistently outperform zero-shot LLMs, highlighting the importance of task-specific adaptation in legal NLP applications.

Future studies could apply the Criticality Prediction task in other legal contexts by incorporating sources from different jurisdictions and languages. This approach would broaden the research's impact and enhance the model's adaptability across different legal systems.

Limitations

It is very difficult to estimate the importance of a case. By relying on proxies such as whether the case was converted to a leading decision (LD-label) and how often this leading decision was cited (Citation-label), we were able to create labels algorithmically. While we discussed this with lawyers at length and implemented the solution we agreed on finally, this task remains somewhat artificial. Additionally, it is worth noting that LLMs can be sensitive to prompt formats and the order of answer options, leading to inconsistent outputs as highlighted in recent research (Webson and Pavlick, 2022; Elazar et al., 2021). Incorporating varied prompt designs into future studies could strengthen our approach further.

Ethics Statement

While automating case prioritization and identifying leading decisions can greatly benefit legal professionals, there are potential risks associated with deploying such classifiers. One concern is the risk of perpetuating biases present in historical legal trends. For instance, case prioritization decisions should not be influenced by factors such as gender, race, or other protected characteristics. We acknowledge these concerns and will pursue measures to mitigate such biases in future work.

Additionally, there are challenges with reproducibility when using closed models like ChatGPT. Since the internal workings of these models are not fully transparent, results may be difficult to replicate. To promote open science, we have provided comprehensive evaluations of open source multilingual models, aiming to make our findings more accessible and reproducible.

Acknowledgments

We thank Luca Rolshoven and Mara Häusler for their thoughtful feedback on the manuscript, and the anonymous reviewers for their insightful and constructive comments.

References

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective. *PeerJ Computer Science*, 2:e93. Publisher: PeerJ Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Ilias Chalkidis. 2023. Chatgpt may pass the bar exam soon, but has a long way to go for the lexglue benchmark. *Preprint*, arXiv:2304.12202.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural Legal Judgment Prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Yi Feng, Chuanyi Li, and Vincent Ng. 2022. Legal judgment prediction: A survey of the state of the art. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*,

pages 5461–5469. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Maria Medvedeva, Michel Vols, and Martijn Wieling. 2018. Judicial decisions of the European Court of Human Rights: looking into the crystall ball. *Proceedings of the Conference on Empirical Legal Studies in Europe 2018*.

Joel Niklaus, Ilias Chalkidis, and Matthias Stürmer. 2021. Swiss-Judgment-Prediction: A Multilingual Legal Judgment Prediction Benchmark. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 19–35, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Joel Niklaus, Veton Matoshi, Pooja Rani, Andrea Galassi, Matthias Stürmer, and Ilias Chalkidis. 2023. LEXTREME: A multi-lingual and multi-task benchmark for the legal domain. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3016–3054, Singapore. Association for Computational Linguistics.

Joel Niklaus, Matthias Stürmer, and Ilias Chalkidis. 2022. An Empirical Study on Cross-X Transfer for Legal Judgment Prediction. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 32–46, Online only. Association for Computational Linguistics.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the Curse of Multilinguality by Pre-training Modular Transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States. Association for Computational Linguistics.

Vishvaksenan Rasiah, Ronja Stern, Veton Matoshi, Matthias Sturmer, Ilias Chalkidis, Daniel Ho, and Joel Niklaus. 2023. One law, many languages: Benchmarking multilingual legal reasoning for judicial support.

V Sanh. 2019. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M. Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet

Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Karen Fort, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A. Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S. Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2022. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *arXiv preprint*. ArXiv:2211.05100 [cs].

Gil Semo, Dor Bernsohn, Ben Hagag, Gila Hayat, and Joel Niklaus. 2022. ClassActionPrediction: A Challenging Benchmark for Legal Judgment Prediction of Class Action Cases in the US. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 31–46, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. We Need to Talk About Random Splits. *arXiv:2005.00636 [cs]*. ArXiv: 2005.00636.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint*. ArXiv:2302.13971 [cs].

Jannis Vamvas, Johannes Graën, and Rico Sennrich. 2023. SwissBERT: The Multilingual Language Model for Switzerland. *arXiv e-prints*, pages arXiv–2303.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv:2010.11934 [cs]*. ArXiv: 2010.11934.

Octavia-Maria Şulea, Marcos Zampieri, Mihaela Vela, and Josef van Genabith. 2017. Predicting the Law Area and Decisions of French Supreme Court Cases. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 716–722, Varna, Bulgaria. INCOMA Ltd.

A Weighting Formula for Citation-Label

The weighting formula used for the **Citation-Label** is designed to balance the impact of older cases with more recent rulings, ensuring that the ranking reflects both citation frequency and recency. The formula is as follows:

$$score = count \times \frac{year - 2002 + 1}{2023 - 2002 + 1}$$

Where:

- *count* refers to the number of times a particular case is cited in Swiss Federal Supreme Court (SFSC) decisions.
- The year 2002 is the starting year of our dataset.
- The year 2023 is the end point of the our dataset.

We have +1 adjustment in weighting factor $\frac{year-2002+1}{2023-2002+1}$. This ensures that cases from the year 2002 are still included in the weighted calculation and do not receive a weight of zero.

We did not use more than four bins for the Citation-Label to ensure enough examples also for the smallest class while still adding more granularity compared to the LD-Label.

B Dataset Example

Figure 3 shows an example input, target and corresponding metadata.

C General Dataset Metadata

Table 4 shows metadata for cantons, courts, chambers law-areas and languages.

D Dataset Licensing

The original case data is available from the Swiss Federal Supreme Court[2] and the Entscheidsuche portal[3] was used to download HTML files for each case.

In compliance with the Swiss Federal Supreme Court's licensing policy[4], we are releasing the dataset under a CC-BY-4.0 license. *The link to the dataset will be made available upon acceptance.*

Personally identifying information has already been anonymized by the Swiss Federal Supreme Court in accordance with its anonymization rules[5].

E Zero-shot Prompts

Figures 4 and 5 show the zero-shot prompts we used for the Case Prioritization and Leading Decision Selection tasks respectively.

F Hyperparameters and Package Settings

We used a fixed learning rate of 1e-5 without tuning, running each experiment with three random seeds (1-3) and excluding seeds with high evaluation losses. Gradient accumulation was applied when GPU memory was insufficient to maintain a final batch size of 64. Training employed early stopping with a patience of 5 epochs, based on validation loss. To reduce costs, AMP mixed precision was used where it didn't cause overflows (e.g., mDeBERTa-v3). The max-sequence length was set at 2048 for Facts and 4096 for Considerations. For LLM evaluations, we used the ChatCompletion API for GPT-3.5 (as of June 7, 2023), and ran LLaMA-2 locally with 4-bit quantization.

For the analysis of consideration and fact lengths, we used SpaCy's *en_core_web_sm* for tokenization.

G Resources

The experiments were run on NVIDIA GPUs, including the 24GB RTX3090, 32GB V100, 48GB A6000, and 80GB A100, using approximately 50 GPU days in total.

H Additional Results

Table 5 shows results with standard deviations on the test set. Table 6 shows scores on the validation set.

I Use of AI Assistants

We used ChatGPT and Claude to enhance grammatical correctness and style, and utilized Google Colab's Generate AI feature for some of the dataset analysis.

[2] https://www.bger.ch/de/index.htm

[3] https://entscheidsuche.ch/

[4] https://www.bger.ch/files/live/sites/bger/files/pdf/de/urteilsveroeffentlichung_d.pdf

[5] https://www.bger.ch/files/live/sites/bger/files/pdf/Reglemente/Anonymisierungsregeln_2020_def__d.pdf

Input
[Consideraions]: Erwägungen: 1. Angefochten ist der in einem kantonal letztinstanzlichen Scheidungsurteil festgesetzte nacheheliche Unterhalt in einem Fr. 30'000.– übersteigenden Umfang; auf die Beschwerde ist somit einzutreten (Art. 72 Abs. 1, Art. 74 Abs. 1 lit. b, Art. 75 Abs. 1 und Art. 90 BGG). 2. Die Parteien pflegten eine klassische Rollenteilung, bei der die Ehefrau die Kinder grosszog und sich um den Haushalt kümmerte. Infolge der Trennung nahm sie im November 2005 wieder eine Arbeitstätigkeit auf und erzielt mit einem 80%-Pensum Fr. 2'955.– netto pro Monat. Beide kantonalen Instanzen haben ihr jedoch auf der Basis einer Vollzeitstelle ein hypothetisches Einkommen von Fr. 3'690.– angerechnet. Das Obergericht hat zwar festgehalten, der Ehefrau sei eine Ausdehnung der Arbeitstätigkeit kaum möglich, gleichzeitig aber erwogen, es sei nicht ersichtlich, weshalb sie nicht einer Vollzeitbeschäftigung nachgehen könne. Ungeachtet dieses Widerspruches wird das Einkommen von Fr. 3'690.– von der Ehefrau ausdrücklich anerkannt, weshalb den nachfolgenden rechtlichen Ausführungen dieser Betrag zugrunde zu legen ist. [...]

Metadata:
Decision ID: *65aad3f6-33c2-4de2-91c7-436e8143d6ea*
Year: *2007*
Language: *German*
Law Area: *Civil*
LD Label: *Critical*
Citation Label: *Citation-1*
Court: *CH_BGer*
Chamber: *CH_BGer_005,*
Canton: *CH*
Region: *Federation*

Target:
critical-1
Possible LD label: critical, non-critical, Possible citation label: critical-1, critical-2, critical-3, critical-4

Figure 3: Example of an input, target and corresponding metadata

Table 4: Metadata for cantons, courts, chambers, law-areas, and languages

Metadata	Number	Examples
Cantons	1	Federation (CH)
Courts	1	Supreme Court
Chambers	13	CH-BGer-011 CH-BGer-004 CH-BGer-008 CH-BGer-002 CH-BGer-005 CH-BGer-001 CH-BGer-006 CH-BGer-009 CH-BGer-015 ...
Law-Areas	4	Civil, Criminal, Public, Social
Languages	5	German, French, Italian

Criticality Prediction (CP) LD Facts/Consideration

Given the **{facts/considerations}** from the following Swiss Federal Supreme Court Decision:
{INPUT FROM THE VALIDATION SET}
Federal Supreme Court Decisions in Switzerland that are published additionally get the label critical, those Federal Supreme Court Decisions that are not published additionally, get the label non-critical. Therefore, there are two labels to choose from:
- **critical**
- **non-critical**
The relevant label in this case is:

Figure 4: Prompt used for Criticality Prediction (LD Facts/-Consideration). The LLM is tasked with predicting whether a Swiss Federal Supreme Court decision is labeled as critical or non-critical based on the provided facts or considerations

Table 5: Configuration aggregate scores with standard deviations on the test set. The macro-F1 scores are provided.

Model	LD-F	LD-C	C-F	C-C	Agg.
MiniLM	$54.7_{+/-1.9}$	$65.8_{+/-1.6}$	$9.8_{+/-2.8}$	$20.8_{+/-3.0}$	21.8
DistilmBERT	$56.2_{+/-0.5}$	$65.4_{+/-1.7}$	$19.6_{+/-1.1}$	$22.1_{+/-0.4}$	30.9
mDeBERTa-v3	$55.1_{+/-2.0}$	$69.8_{+/-2.8}$	$21.0_{+/-3.6}$	$17.5_{+/-4.4}$	29.1
XLM-R_{Base}	$57.2_{+/-1.5}$	$65.9_{+/-3.2}$	$21.3_{+/-1.5}$	$23.7_{+/-1.9}$	32.8
XLM-R_{Large}	$56.4_{+/-1.8}$	$67.9_{+/-1.9}$	$24.4_{+/-7.2}$	$29.1_{+/-2.7}$	37.1
X-MOD_{Base}	$56.6_{+/-1.8}$	$67.8_{+/-2.9}$	$20.0_{+/-3.0}$	$20.6_{+/-3.5}$	30.5
$SwissBERT_{(xlm\text{-}vocab)}$	$56.9_{+/-0.7}$	$67.3_{+/-4.7}$	$25.7_{+/-8.3}$	$23.0_{+/-4.0}$	34.8
$mT5_{Small}$	$52.2_{+/-1.9}$	$62.1_{+/-5.2}$	$13.2_{+/-2.4}$	$17.9_{+/-1.7}$	24.0
$mT5_{Base}$	$52.1_{+/-1.6}$	$61.5_{+/-3.9}$	$14.0_{+/-2.8}$	$19.7_{+/-1.6}$	25.4
BLOOM-560m	$53.0_{+/-1.7}$	$61.7_{+/-4.1}$	$10.7_{+/-3.7}$	$8.0_{+/-3.5}$	15.8
Legal-Swiss-R_{Base}	$57.7_{+/-1.6}$	$70.5_{+/-2.3}$	$16.2_{+/-5.8}$	$20.1_{+/-5.6}$	28.0
Legal-Swiss-R_{Large}	$55.9_{+/-2.2}$	$68.9_{+/-2.1}$	$25.8_{+/-7.8}$	$16.3_{+/-8.7}$	30.2
Legal-Swiss-LF_{Base}	$58.1_{+/-2.1}$	$70.8_{+/-2.9}$	$21.4_{+/-2.9}$	$17.4_{+/-8.6}$	29.5

Criticality Prediction (CP) Citation Facts/Consideration
Given the **{facts/considerations}** from the following Swiss Federal Supreme Court Decision:
{INPUT FROM THE VALIDATION SET}
How likely is it that this Swiss Federal Supreme Court Decision gets cited. Choose between one of the following labels (a bigger number in the label means that the court decision is more likely to be cited):
- **critical-1**
- **critical-2**
- **critical-3**
- **critical-4**
The relevant label in this case is:

Figure 5: Prompt used for Criticality Prediction (Citation Facts/Consideration). The LLM is tasked with predicting how likely a Swiss Federal Supreme Court decision is to be cited, using a four-tiered label (critical-1 to critical-4), based on the provided facts or considerations.

Table 6: Configuration aggregate scores on the validation set. The macro-F1 scores are provided. The highest values are in bold. It is important to note that the scores presented here are calculated as the harmonic mean over multiple seeds.

Model	LD-F	LD-C	C-F	C-C	Agg.
MiniLM	59.1	71.0	14.9	36.9	31.9
DistilmBERT	59.6	70.1	26.3	35.8	41.2
mDeBERTa-v3	60.1	73.0	30.4	36.0	44.0
XLM-R_{Base}	60.1	70.5	26.9	38.5	42.6
XLM-R_{Large}	60.5	71.7	27.2	39.7	43.3
X-MOD_{Base}	57.1	71.0	27.0	33.4	40.6
$SwissBERT_{(xlm\text{-}vocab)}$	59.0	72.1	29.4	38.8	44.1
$mT5_{Small}$	54.8	66.1	26.3	32.5	39.2
$mT5_{Base}$	55.7	64.4	24.3	29.3	36.8
BLOOM-560m	52.2	64.3	20.1	21.8	30.7
Legal-Swiss-R_{Base}	61.2	**73.6**	27.7	41.0	44.2
Legal-Swiss-R_{Large}	**61.8**	73.5	29.8	32.0	42.3
Legal-Swiss-LF_{Base}	59.4	72.7	**32.2**	**42.5**	**47.0**

Revisiting LLMs as Zero-Shot Time-Series Forecasters: Small Noise Can Break Large Models

Junwoo Park[1,2], **Hyuck Lee**[2], **Dohyun Lee**[1,2],
Daehoon Gwak[1], **Jaegul Choo**[1],
[1]KAIST AI [2]KRAFTON
{junwoo.park,aiclaudev,daehoon.gwak,jchoo}@kaist.ac.kr
dlgur0921@krafton.com

Abstract

Large Language Models (LLMs) have shown remarkable performance across diverse tasks without domain-specific training, fueling interest in their potential for time-series forecasting. While LLMs have shown potential in zero-shot forecasting through prompting alone, recent studies suggest that LLMs lack inherent effectiveness in forecasting. Given these conflicting findings, a rigorous validation is essential for drawing reliable conclusions. In this paper, we evaluate the effectiveness of LLMs as zero-shot forecasters compared to state-of-the-art domain-specific models. Our experiments show that LLM-based zero-shot forecasters often struggle to achieve high accuracy due to their sensitivity to noise, underperforming even simple domain-specific models. We have explored solutions to reduce LLMs' sensitivity to noise in the zero-shot setting, but improving their robustness remains a significant challenge. Our findings suggest that rather than emphasizing zero-shot forecasting, a more promising direction would be to focus on fine-tuning LLMs to better process numerical sequences. Our experimental code is available at https://github.com/junwoopark92/revisiting-LLMs-zeroshot-forecaster

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in diverse tasks, even without domain-specific training (Kojima et al., 2022; Wei et al., 2022). Building on this success, researchers are increasingly leveraging pre-trained LLMs for time series forecasting (Zhou et al., 2023; Jin et al., 2023; Liu et al., 2024a; Gruver et al., 2023; Requeima et al., 2024; Tang et al., 2025). A common approach involves converting the time series data into a format that LLMs can process, such as through alignment-tuning or by encoding sequences directly within prompts. This enables LLMs to perform few-shot or zero-shot forecasting. These studies show that LLM-based forecasters consistently perform well across multiple domains while reducing the need for domain-specific training. Notably, LLMTime (Gruver et al., 2023) demonstrated that LLMs can achieve zero-shot time series forecasting, performing at a level comparable to or even surpassing domain-specific models trained on the full dataset.

However, there remains an ongoing discussion about whether a thorough evaluation was conducted when integrating LLMs into time-series forecasting and assessing their performance. For instance, existing studies have used domain-specific models (Zhou et al., 2021, 2022; Wu et al., 2021) as baselines for comparison with LLMs, but these baseline models may not be sufficiently powerful (Zeng et al., 2023) to provide a fair assessment of LLM performance. Furthermore, a recent comprehensive study (Tan et al., 2024) found that LLMs do not significantly contribute to forecasting performance, undermining the justification for their high computational cost. These concerns discourage the adoption of LLMs for promising applications in time-series forecasting. Therefore, it is crucial to reassess the true performance of LLM-based forecasters. To address this, we raise the following question:

Are LLMs as zero-shot forecasters truly more effective than domain-specific models?

Typically, zero-shot forecasting is useful because it does not require domain-specific training. However, if inference with a zero-shot model takes too much time, the end-to-end process of training and deploying a domain-specific model could be faster, which would undermine the main advantage of using a zero-shot approach. Therefore, to be considered a viable alternative, a zero-shot forecaster must at least offer faster inference than the total time required to train and deploy a domain-specific

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 906–922
July 27 - August 1, 2025

model. While inference speed is essential, accuracy remains equally important in scenarios where real-time prediction is not required. Domain-specific models often rely on high-quality training data, and their performance can be degraded in new domains with limited data. In such cases, pre-trained LLMs may offer better generalization. Therefore, even if an LLM is slower, a higher forecast accuracy can justify its use, making precision a valid criterion for assessing viability.

In summary, for LLMs to be viable zero-shot forecasters, they should meet at least one of two key criteria. *(1) they should generate predictions faster than domain-specific models that need to be trained for each new domain*, and *(2) their forecasting accuracy should be superior to, if not at least comparable to, domain-specific models*. Even if only the first criterion is satisfied, meaning the model generates predictions faster than domain-specific models including their training time, it is still suitable for applications where real-time inference is important. Even if only the second criterion is satisfied, meaning the model achieves forecasting accuracy comparable to or better than domain-specific models, it remains useful in cases where performance is critical but immediate inference is not required.

In this paper, we thoroughly evaluate the effectiveness of LLM-based zero-shot forecasters by comparing their inference speed and accuracy against state-of-the-art (SoTA) domain-specific forecasting models. Our experimental results indicate that LLM forecasters struggle to match the forecasting accuracy of domain-specific models. Furthermore, linear models, which can be trained on single input sequence, not only outperformed LLM-based forecasters in forecasting accuracy but also are more cost-efficient. Additionally, our analysis reveals that the low accuracy of LLMs in time-series forecasting is closely tied to the presence of noise commonly found in real-world datasets. Despite exploring various strategies to mitigate this issue in zero-shot settings, our experiments show only marginal improvements, highlighting noise sensitivity as a fundamental limitation of LLM-based forecasters.

2 Experimental Setup

Problem Definition Time series forecasting is the task of predicting future values based solely on past observations. Formally, given a uniformly sampled n observations of time series s in which each observation has d dimensions, $s_n(t) = \{x_t, x_{t+1}, \ldots, x_{t+n-1}\}, \quad x \in \mathbb{R}^d$, a forecaster predicts the future sequence $s_O(t+I)$ based on $s_I(t)$ where I and O denote an input length and output length of sequence, respectively. We define the application of a forecaster across multiple domains as follows: Given the N datasets $\{D^1, \ldots, D^N\}$, a forecaster predicts futures based on the most recent input sequence s_I^i of each D^i. To solve this problem, we introduce three approaches: prompt-based zero-shot forecasting, domain-specific forecasting, and single-shot forecasting.

Prompt-based LLM Forecasting Methods Prompt-based zero-shot forecasting involves converting s_I into a text sequence within a prompt and then asking LLMs to continue the sequence up to the desired length O. Existing methods, such as LLMTime and LLMP (Requeima et al., 2024), provide a foundation for LLM-based forecasting, but exploring additional prompting strategies can offer deeper insights into LLM capabilities. To this end, we designed TS-CoT and TS-InContext prompts (Figure 6), inspired by Chain of Thought (CoT) and In-Context Learning. We use both black-box (GPT-3.5, GPT-4, GPT-4o) and white-box models (LLaMA 2-7B, 3-8B, 2-70B, 3-70B, 3.1-70B).

Domain-Specific Forecasting Models Gruver et al. (2023) demonstrated that LLMTime outperforms conventional forecasting methods trained on the entire dataset, despite relying solely on the input sequence for prediction. However, most baseline models (Wu et al., 2021; Zhou et al., 2022) are regarded as ineffective baselines (Zeng et al., 2023) and are relatively outdated compared to recently proposed forecasting models. Therefore, to assess whether utilizing LLMs for forecasting is genuinely meaningful, we conduct comparisons with recent SoTA forecasting models (Wu et al., 2023; Nie et al., 2023; Liu et al., 2024b; Wang et al., 2024). The characteristics of each model are detailed in Appendix A.1.

Domain-Specific Single-shot Linear Models Additionally, the linear models (Zeng et al., 2023) have been shown to achieve strong performance on widely used benchmarks despite having a small number of parameters. We found that these models can be trained on the same amount of data as LLMTime receives as input. Thus, we train linear models with only one input sequence and include them as single-shot linear models (details in Appendix B.3).

Evaluation Protocol To evaluate computational

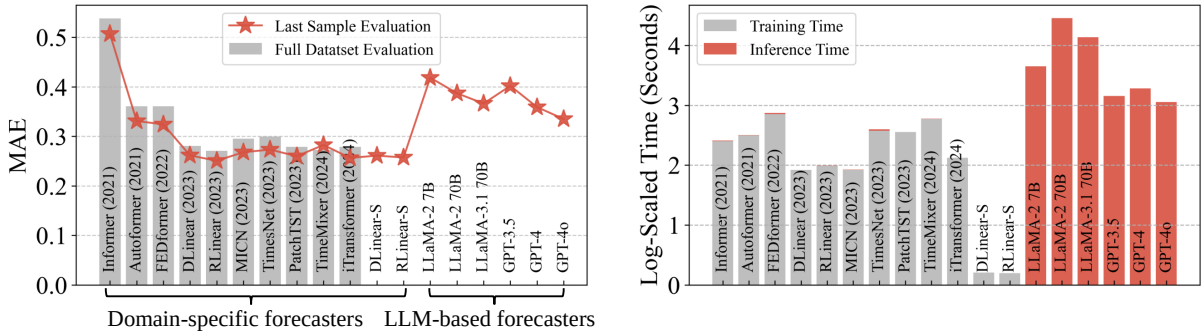


Figure 1: Multivariate forecasting results on five datasets from the Informer benchmark (I=$\{96, 384\}$ and O=$\{48, 192\}$). We report the averaged MAE and log-scaled inference time for domain-specific forecasters and LLM-based forecasters using the best-performing LLMTime prompt (results for other three prompts are provided in Table 5). (Left) Recent domain-specific forecasters achieve lower MAE than LLM-based forecasters in last sample evaluation (Gruver et al., 2023). (Right) Overall, LLM-based forecasters have longer inference times than domain-specific models, despite the latter requiring domain-specific training before inference. Moreover, single-shot linear models (DLinear-S and RLinear-S), trained solely on the input sequence of each domain, achieve significantly shorter inference times and outperform LLM-based forecasters. Appendix B.3 includes a detailed evaluation protocol. Figure 7 shows qualitative results and Figure 12 includes full results with confidence intervals.

cost, we define the total computation cost C of each approach as the sum of training cost C^{T} and inference cost C^{I}. We assume that domain-specific models require separate training and inference for each dataset D^i, leading to $C_{\text{Domain}} = \frac{1}{N}\sum_{i=1}^{N}\left[C^{\mathrm{T}}(D^i) + C^{\mathrm{I}}(s_I^i)\right]$. For pre-trained LLMs, we assume no additional training cost, so only the inference cost accumulates: $C_{\text{LLM}} = \frac{1}{N}\sum_{i=1}^{N} C^{\mathrm{I}}(s_I^i)$. Finally, the single-shot linear model also trained with a input sequence s^i in each dataset independently before inference, resulting in $C_{\text{Linear}} = \frac{1}{N}\sum_{i=1}^{N}\left[C^{\mathrm{T}}(\{s_I^i\}) + C^{\mathrm{I}}(s_I^i)\right]$. We consider LLM forecasters cost-efficient as zero-shot forecasters only if $C_{\text{LLM}} < C_{\text{domain}}$ and $C_{\text{LLM}} < C_{\text{Linear}}$. Appendix B.3 shows details for calculation of the inference time. We measured the mean absolute error (MAE) and the mean squared error (MSE) to evaluate the forecasts.

Benchmark Datasets We evaluate all forecasters using three benchmark datasets: Monash (Godahewa et al., 2021), and Function (Gruver et al., 2023), Informer (Zhou et al., 2021) datasets. The Monash dataset consists of real-world time series from various domains. The Function dataset contains numerical sequences from mathematical functions to test pattern learning and extrapolation. The Informer dataset includes multivariate time-series data for long-sequence forecasting, having larger test set than others. Thus, to reduce LLM costs, we evaluate performance using only the last test sample. Appendix B.1 provide details of the datasets.

3 Comparison with SoTA Domain-Specific Forecasters

Finding 1: *SoTA domain-specific models are more accurate and cost-efficient than prompt-based LLM forecasters.*

We compare LLM-based forecasting models with SoTA domain-specific models (TimeMixer and iTransformer) in terms of computational efficiency and accuracy. As shown in Figure 1, LLMTime with LLaMA-3.1-70B performs comparably to forecasting models such as Autoformer and FEDformer, as evaluated in their original paper. However, it falls short compared to more recent SoTA models. Notably, LLM forecasters underperform compared to single-shot linear models (DLinear-S and RLinear-S), despite these linear models being trained on just a single input sequence. Moreover, these linear models achieve significantly shorter inference time, even including training time, compared to LLMTime with LLaMA-2-7B.

Due to space constraints, we provide additional experimental results in Appendix C.2. Appendix C.2.1 presents an analysis of how different decoding settings (*e.g.*, temperature and top-p values) affect forecasting performance across various configurations, as summarized in Tables 6 and 7. Appendix C.2.2 extends the evaluation beyond the last-sample setting to cover the entire temporal horizon in univariate forecasting, as shown in Table 8. Finally, Appendix C.2.4 pro-

Models	Clean		Gaussian		Constant		Missing	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
GPT-3.5	0.8247	1.4464	0.9215	1.6484	0.9018	1.4923	0.9897	1.7561
LLAMA2 7B	0.8875	1.5366	0.8831	1.4087	0.9380	1.6075	0.9482	1.4286
LLAMA3 8B	0.7205	1.0185	1.0307	1.9737	0.8974	1.4575	1.0004	1.5850
GPT-4	**0.6956**	0.9533	0.7330	1.0234	0.7851	1.1315	0.9115	1.4298
LLAMA2 70B	0.7196	1.0028	0.7776	1.2476	0.8773	1.3353	0.8404	1.2749
LLAMA3 70B	0.7006	0.9596	0.8870	1.5695	0.8184	1.1474	0.8461	1.1829
DLinear-S	0.7074	**0.8890**	**0.7063**	**0.8896**	**0.7120**	**0.8997**	**0.7112**	**0.8910**
RLinear-S	0.7182	0.8804	0.7196	0.8817	0.7171	0.8766	0.7438	0.8971

Table 1: When the Monash dataset contains Gaussian, Constant, and Missing noise, commonly used to evaluate noise robustness (Cheng et al., 2024), LLM-based forecasters experience a performance decline, while Linear-S models perform similarly to clean data.

vides additional baseline comparisons, including results for ARIMA (Box and Jenkins, 1968) and N-BEATS (Oreshkin et al., 2019), alongside the linear models.

3.1 Origin of LLMs Performance Degradation

Finding 2: *LLMs are highly sensitive to noise, while linear models remain robust.*

Another point of inquiry stems from the fact that LLMs have clearly demonstrated near-perfect performance on various functional datasets and accurate prediction results in the Monash dataset (Gruver et al., 2023). In reproducing these experiments, we discovered that these datasets, unlike Informer datasets, contain very little noise and are clean.

Qualitative Results on the Function Dataset For the function dataset, Figure 2 shows how the performance of LLMs changes when a Gaussian noise is added to the original data. Even a very small amount of noise, relative to the original scale, can lead to significantly larger errors than the original predictions. This finding suggests that LLMs are highly sensitive to noise.

Quantitative Results on the Monash Dataset We also compare LLMs with single-shot linear models using the Monash dataset across various noise types, which are typically addressed in the noise robustness literature. Table 1 shows that LLM-based methods significantly decline in performance compared to their original performance when exposed to three types of noise (details in Appendix B.4). On the other hand, the linear models maintain nearly the same level of performance even under various types of noise. This result aligns with the theoretical findings of Cheng et al. (2024), which suggest that models optimized with L1 and L2 loss exhibit robustness to noise. In addition to the three representative noise types discussed here, further evaluations under additional noise conditions—such as frequency-based noise—are presented in Appendix C.2.3.

Models	Filtering	ETTm2		Exchange Rate		Weather	
		MAE	MSE	MAE	MSE	MAE	MSE
GPT-4o	X	0.2322	0.1237	0.3595	0.2893	0.1880	0.1147
GPT-4o	Gaussian	0.1995	0.1097	0.3718	0.2643	0.1698	0.0827
GPT-4o	EMA	0.2501	0.1519	0.3615	0.2882	0.1808	0.0972
LLAMA3.1 70B	X	0.2462	0.1579	0.4646	0.3596	0.1659	0.0741
LLAMA3.1 70B	Gaussian	0.2659	0.1868	0.4323	0.3579	0.1735	0.0981
LLAMA3.1 70B	EMA	0.2431	0.1661	0.3765	0.2782	0.1689	0.0955
DLinear-S	X	0.2039	0.0782	**0.2296**	**0.1166**	0.1643	0.0522
RLinear-S	X	**0.1824**	**0.0677**	0.2571	0.1370	**0.1392**	**0.0469**

Table 2: When two noise filtering methods (Gaussian and EMA) are applied to LLM-based models on the Informer dataset, performance improves (highlighted in red). Nevertheless, single-shot linear models consistently outperform LLM-based models.

3.2 Enhancing Noise Robustness of LLMs

Finding 3: *Additional input samples and noise filtering improve noise robustness of LLMs, but limited.*

Increasing Input Length to Improve Noise Robustness Because the robustness of conventional time-series models to noise depends on the number of training samples, we investigated whether providing longer input sequences could enhance the resilience of LLMs to noise. We hypothesized that using short input sequences might limit LLMs' ability to distinguish noise; therefore, we evaluated their performance by gradually increasing the sequence length. As illustrated in Figure 3, where the period in the Fourier series corresponds to an input sequence length, LLMs typically show only slight improvements in performance, whereas linear models gain significantly from longer sequences.

What If Applying Noise Filtering? The straightforward way to deal with noise is to remove it as a pre-processing for input. To observe how they affect performance, we applied two noise filtering techniques—Gaussian and Exponential Moving Average (EMA) filtering—on real-world datasets (details in Appendix B.4). As shown in Table 2, even after applying noise filtering techniques, LLM performance improves only slightly or remains unchanged.

4 Discussion

In this paper, we empirically demonstrate the limitations of LLM-based forecasting in real-world scenarios where noise is prevalent. A potential reason for LLMs' susceptibility to noise lies in their token-

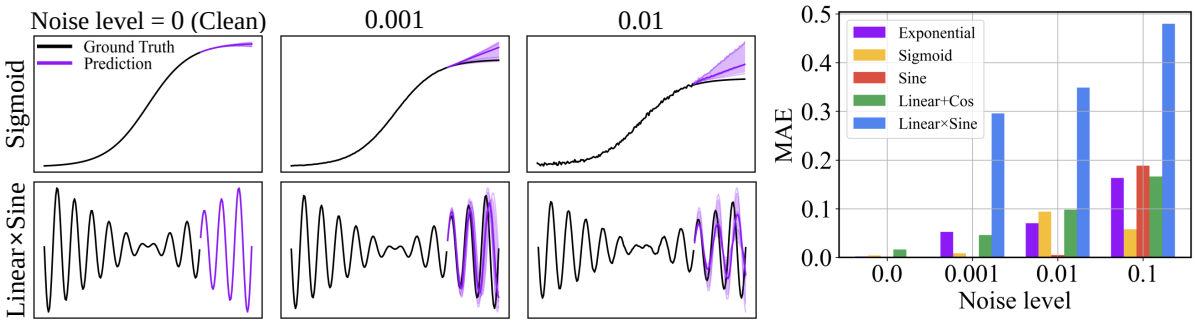


Figure 2: Performance variations in the Function dataset based on noise levels. LLMTime (GPT-4) performed perfectly on clean data, but even a slight noise (*e.g.,* Gaussian noise with a standard deviation of 0.001) addition within the 0-1 input scale significantly increased MAE.

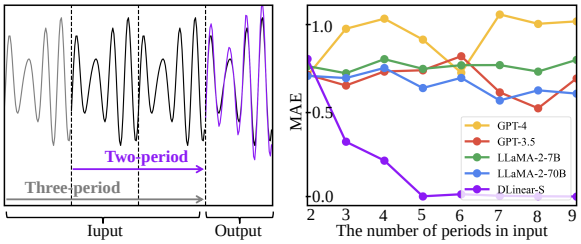


Figure 3: Increasing the input sequence length in prompts for LLM-based models leads to a slight performance improvement, but the gains remain minimal compared to DLinear-S.

based encoding, which amplifies distortions in representation rather than reflecting actual numerical differences. This distortions hinder their ability to recognize meaningful patterns, making extrapolation more challenging. Given the strong reasoning capabilities of LLMs, we argue that rather than emphasizing zero-shot forecasting, a more promising research direction would be enhancing the capability to process numerical sequences through fine-tuning (Hu et al., 2022; Zha et al., 2022; Dettmers et al., 2023).

5 Limitations

Our study provides a comprehensive evaluation of LLM-based zero-shot forecasting, but there are areas that warrant further exploration. While we identify the sensitivity of LLMs to noise, the exact mechanisms behind this issue, such as the impact of tokenization and encoding, deserve deeper investigation. Our evaluation prioritizes computational cost and accuracy, yet aspects like interpretability, adaptability to diverse domains, and the potential benefits of integrating textual or multimodal data could offer valuable insights for future studies. Additionally, our experiments cover widely used benchmark datasets, but assessing LLM performance across an even broader range of real-world forecasting tasks would further strengthen our understanding of their capabilities and limitations.

Acknowledgments

This work was supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190075, Artificial Intelligence Graduate School Program at KAIST) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-00555621).

References

George EP Box and Gwilym M Jenkins. 1968. Some recent advances in forecasting and control. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 17(2):91–109.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hao Cheng, Qingsong Wen, Yang Liu, and Liang Sun. 2024. Robusttsf: Towards theory and design of robust time series forecasting with anomalies. In *Proc. the International Conference on Learning Representations (ICLR)*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I Webb, Rob J Hyndman, and Pablo Montero-Manso. 2021. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. 2023. Large language models are zero-shot time series forecasters. In *Proc. the Advances in*

Neural Information Processing Systems (NeurIPS), volume 36.

Daehoon Gwak, Junwoo Park, Minho Park, Chaehun Park, Hyunchan Lee, Edward Choi, and Jaegul Choo. 2024. Forecasting future international events: A reliable dataset for text-based event modeling. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proc. the International Conference on Learning Representations (ICLR)*.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*.

Peiyuan Liu, H Guo, T Dai, N Li, J Bao, X Ren, Y Jiang, and ST Xia. 2024a. Calf: Aligning llms for time series forecasting via cross-modal fine-tuning. *arXiv preprint arXiv:2403.07300*.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024b. itransformer: Inverted transformers are effective for time series forecasting. In *Proc. the International Conference on Learning Representations (ICLR)*.

Minh Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2024. Turning up the heat: Min-p sampling for creative and coherent llm outputs. *arXiv preprint arXiv:2407.01082*.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A time series is worth 64 words: Long-term forecasting with transformers. In *Proc. the International Conference on Learning Representations (ICLR)*.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2019. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *Proc. the International Conference on Learning Representations (ICLR)*.

Junwoo Park, Daehoon Gwak, Jaegul Choo, and Edward Choi. 2024. Self-supervised contrastive learning for long-term forecasting. In *Proc. the International Conference on Learning Representations (ICLR)*.

Junwoo Park, Jungsoo Lee, Youngin Cho, Woncheol Shin, Dongmin Kim, Jaegul Choo, and Edward Choi. 2023. Deep imbalanced time-series forecasting via local discrepancy density. In *Proc. of Machine Learning and Knowledge Discovery in Databases: Research Track (ECML/PKDD)*, pages 139–155. Springer Nature Switzerland.

Xiangyu Peng, Congying Xia, Xinyi Yang, Caiming Xiong, Chien-Sheng Wu, and Chen Xing. 2024. Regenesis: LLMs can grow into reasoning generalists via self-improvement. *arXiv preprint arXiv:2410.02108*.

James Requeima, John Bronskill, Dami Choi, Richard E Turner, and David Duvenaud. 2024. Llm processes: Numerical predictive distributions conditioned on natural language. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*.

Mingtian Tan, Mike A Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. 2024. Are language models actually useful for time series forecasting? In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*.

Hua Tang, Chong Zhang, Mingyu Jin, Qinkai Yu, Zhenting Wang, Xiaobo Jin, Yongfeng Zhang, and Mengnan Du. 2025. Time series forecasting with llms: Understanding and enhancing model capabilities. *ACM SIGKDD Explorations Newsletter*.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. 2023. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *Proc. the International Conference on Learning Representations (ICLR)*.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. 2024. Timemixer: Decomposable multiscale mixing for time series forecasting. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, et al. 2024. Context is key: A benchmark for forecasting with essential textual information. *arXiv preprint arXiv:2410.18959*.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *Proc. the International Conference on Learning Representations (ICLR)*.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*.

Zhijian Xu, Ailing Zeng, and Qiang Xu. 2024. Fits: Modeling time series with $10k$ parameters. In *Proc. the International Conference on Learning Representations (ICLR)*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STar: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In *Proc. the AAAI Conference on Artificial Intelligence (AAAI)*.

Kaiwen Zha, Peng Cao, Yuzhe Yang, and Dina Katabi. 2022. Supervised contrastive regression. *arXiv preprint arXiv:2210.01189*.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proc. the AAAI Conference on Artificial Intelligence (AAAI)*.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proc. the International Conference on Machine Learning (ICML)*.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 36:43322–43355.

Zihao Zhou and Rose Yu. 2025. Can llms understand time series anomalies? In *Proc. the International Conference on Learning Representations (ICLR)*.

A Related Work

A.1 Domain-Specific Models for Time-series Forecasting

Time-series forecasting has evolved from statistical methods to deep learning-based approaches. Informer (Zhou et al., 2021) improved Transformer efficiency with ProbSparse self-attention, reducing computational overhead for long-term forecasting. Early models like LTSF-Linear (Zeng et al., 2023) demonstrated that simple linear regression could outperform complex architectures in structured datasets. PatchTST (Nie et al., 2023) introduced patching techniques inspired by vision models, enabling better locality retention. TimesNet (Wu et al., 2023) leveraged temporal 2D-variation modeling, transforming time series into structured tensors for more efficient feature extraction. iTransformer (Liu et al., 2024b) addressed multivariate dependencies by applying attention mechanisms to variate tokens rather than time steps. TimeMixer (Wang et al., 2024) introduced a fully MLP-based approach, effectively capturing multiscale variations.

Compared to LLM-based methods, these domain-specific models optimize for numerical representation, long-term dependencies (Park et al., 2024), and computational efficiency (Xu et al., 2024). While LLMs process time-series data as token sequences, they lack specialized multiscale decomposition and structured feature extraction found in these tailored models. Informer and PatchTST improve sparsity and efficiency, whereas iTransformer and TimesNet enhance multivariate correlation modeling. Linear models remain relevant, challenging the necessity of deep learning for structured time-series tasks. The progression from linear models to efficient deep learning architectures highlights the importance of designing task-specific solutions for forecasting. Future research should explore hybrid approaches that combine LLM flexibility with time-series-specific optimizations to further improve performance and efficiency.

A.2 LLM-based Time-series Forecasting

The use of LLMs for time series forecasting has recently gained attention, with two primary approaches emerging in the literature. One approach maps numerical sequences into the token embedding space of LLMs with minimal training, treating time series forecasting as a natural extension of next-token prediction. However, recent studies argue that this method does not effectively leverage the capabilities of LLMs for reasoning about time series. Notably, Tan et al. (2024) raise a crucial question: Are Language Models Actually Useful for Time Series Forecasting?. They challenge the assumption that LLM-based models outperform conventional forecasting methods, demonstrating that removing or replacing the LLM component often improves results.

An alternative approach involves transforming numerical sequences into textual representations, allowing LLMs to interpret time series data through natural language prompts. This paradigm is exemplified by LLMTime (Gruver et al., 2023), which was the first to introduce a zero-shot forecasting method using LLMs. Subsequently, LLMP (Requeima et al., 2024) introduced modifications to prompt design, incorporating elements from stochastic processes to enhance predictive accuracy.

Recent studies have increasingly emphasized the importance of textual information in forecasting tasks, demonstrating that integrating external knowledge sources can significantly enhance predictive performance. In particular, LLM-based approaches have shifted towards context-aided forecasting, where models utilize domain-specific textual inputs to inform future predictions. For example, Gwak et al. (2024) show that LLMs can successfully predict future events from past news articles, underscoring their ability to capture temporal and contextual signals from language.

Building on this line of research, the CiK benchmark (Williams et al., 2024) provides a systematic framework for evaluating the role of auxiliary textual information in time-series forecasting. It demonstrates that models incorporating relevant domain knowledge significantly outperform purely numerical baselines. Moreover, CiK proposes evaluation protocols to assess how effectively LLMs integrate multimodal context, including both structured and unstructured inputs. These developments suggest that future research should focus not just on zero-shot capabilities of LLMs, but on methods that enable them to synergistically leverage textual and numerical data for more robust forecasting.

Dataset	Domain	Length	Channels	frequency
ETTm2	System Monitoring	69680	7	15 minutes
Electricity	Energy Consumption	26304	321	1 Hour
Traffic	Traffic	17544	862	1 Hour
Weather	Weather	52695	21	10 Minutes
ExchangeRate	Economic	7588	8	1 Day
Monash Dataset	Diverse domains	[144:900]	8	-
Function Dataset	Mathematical Equations	200	6	-

Table 3: Statistics of Informer, Monash, and Function datasets. In the last sample evaluation, we compare the performance of each model by taking the last $I + O$ length from each dataset. Since LLMs predict each channel independently in multivariate forecasting, the number of sequences that LLMs need to predict is equal to the number of channels.

B Experiment Details

B.1 Dataset Details

To compare LLM forecasters with domain-specific models, we selected three commonly used datasets: Informer (Zhou et al., 2021), Monash (Godahewa et al., 2021), and Function datasets (Gruver et al., 2023). The Informer dataset is a widely used benchmark for evaluating forecasting models, including multivariate time-series data from different domains such as ETTm2, Exchange Rate, Electricity, Traffic and Weather. This dataset mainly focuses on long-sequence forecasting. The Monash consists of public benchmark datasets from the Monash Time Series Forecasting Archive, covering various real-world domains such as energy consumption, traffic, and finance. Among these, we use eight time series from LLMTime (Gruver et al., 2023). Function dataset contains numerical time series sampled from mathematical functions (*e.g.,* sine, sigmoid, beat interference) to assess forecasting models' ability to learn and extrapolate structured numerical patterns.

B.2 Licenses and Terms of Use for Artifacts

We utilize several datasets and pre-trained models, each with specific licensing terms. The Monash Time Series Forecasting (TSF) Repository is licensed under Creative Commons Attribution 4.0 International (CC BY 4.0), allowing research use with proper attribution[1] The Informer datasets do not explicitly state a license, but users are advised to check the official repository for terms of use[2]. The Function dataset similarly lacks stated licensing details, requiring direct consultation with the authors. The study also employs pre-trained models, including GPT-3.5 and GPT-4, which are governed by OpenAI's proprietary API terms, allowing usage under strict conditions[3]. LLaMA models, provided by Meta AI, are subject to a research-focused license, permitting non-commercial research use[4]. Users must comply with each artifact's licensing conditions, particularly regarding research-only restrictions and commercial use limitations.

AI Assistants in Research or Writing We utilized AI assistants, specifically ChatGPT, to support research, writing, and coding. The AI was used for tasks such as drafting text, summarizing information, and refining explanations. This usage is documented to ensure transparency and acknowledgment of AI contributions.

B.3 Details of Comparison with SoTA Domain-Specific Models

Evaluation Protocol A widely used evaluation method for time-series forecasting involves splitting the entire series into training, validation, and test sets while maintaining the chronological order. Specifically, the most recent 20% of the time series is designated as the test set, while the remaining portion is used for training and hyperparameter tuning through validation. Given that time-series forecasting inherently involves predicting future values, it is crucial to split the dataset sequentially to prevent data leakage.

Once the dataset is split into these temporal segments, a sliding window approach is employed, where a fixed-length window moves with a stride through the time series. Each window consists of an input sequence of length I and an output sequence of length O, which serves as the ground truth for evaluating model predictions. The model is trained by minimizing the loss between its predicted output sequence and the actual values. Table 3 provides statistical information about the datasets used, including their total length and the number of samples obtained after applying the sliding window method.

Last Sample Evaluation While this evaluation method is suitable for domain-specific models, it introduces significant computational and financial costs when applied to LLM-based approaches due to the increased number of API calls and processing

[1] https://huggingface.co/datasets/Monash-University/monash_tsf

[2] https://github.com/zhouhaoyi/Informer2020

[3] https://openai.com/terms

[4] https://ai.facebook.com/blog/large-language-model-llama-meta-ai

requirements. To mitigate this burden, LLMTime employs a reduced evaluation strategy, selecting only the last sample from the Informer dataset for testing. Consequently, the number of test samples in each domain is proportional to the number of channels in that domain. Since LLMTime does not disclose the exact methodology used for this selection, our evaluation follows a similar approach by selecting the last available sample from the conventional time-series split. While this approach prevents direct comparison with results from the original paper, we confirm that the general trend of the findings remains consistent.

Domain-Specific Model Details In the field of time-series forecasting, numerous models have emerged since the introduction of Informer, each showcasing strong performance with distinct innovations. We concentrate on evaluating the most advanced and high-performing models from recent developments. Our approach is validated against recent seven forecasting baselines, with all models implemented in PyTorch. For the latest forecasting models—including DLinear[5], MICN[6] (Wang et al., 2023), and PatchTST[7] (Nie et al., 2023), TimesNet (Wu et al., 2023), iTransformer (Liu et al., 2024b), TimeMixer[8] (Wang et al., 2024)—we utilized the official implementations provided by the original authors rather than developing them independently. For domain-specific models, we conducted five independent training and inference runs, varying the random seed in each run.

LLM-based Model Details For LLM-based models, we performed five inference runs on the same test sample and computed the median of the five predictions to mitigate the impact of stochastic variations due to temperature settings. This median aggregation method is commonly used in LLM-based models to reduce the affect of outlier predictions. As a result, domain-specific models report confidence intervals in the results, while LLM-based models do not due to the fivefold increase in computational cost that would be required. The API model utilizes the API provided by OpenAI [9], while all LLaMA models use pretrained models provided by Hugging Face [10].

Training Strategy for Linear-S Models Linear-S models follow a different training and inference process. These models use a single input sequence only for training. To conduct both training and validation with a single sample, we need to create a windowed dataset from the input sequence, similar to when training a domain-specific model. Given an input sequence length I and a target prediction length O, we transform this into a smaller input length I' and output length O' to construct a windowed dataset. The total number of windows K is determined as follows:

$$K = d(I - (I' + O') + 1) \tag{1}$$

where d is the number of channels. The linear model employs a channel independence strategy (Nie et al., 2023), where channels are transformed into the batch axis, allowing for independent predictions across channels. As a result, in multivariate forecasting, the number of windows is proportional to the number of channels. In the sliding window framework, if K is too small, training and validation of the linear model become challenging. Therefore, with I, O, and d fixed, we need to adjust I' and O' to ensure a sufficient number of windowed data points. If O' is smaller than O, inference must be performed in an autoregressive manner, which can lead to accumulated errors and degrade performance. Considering these characteristics, we set I' and O' as half of I and O', respectively. The statistics of each dataset are presented in Table 4.

Informer Dataset	d	I	O	K	I'	O'
ETTm2	7	384	192	1,351	96	96
Exchange Rate	8	384	192	1,544	96	96
Weather	21	384	192	4,053	96	96
Electricity	321	96	48	15,729	24	24
Traffic	862	96	48	42,238	24	24

Table 4: The number of windows K for training and validating the single-shot linear models according to the internal prediction length O' and input length I' for each dataset with the target input I and output O lengths.

Total Inference Time Calculation To compare the inference speeds of domain-specific models and LLM-based models, we measure the total time required for a single prediction. For domain-specific models, this total time includes the training time (utilizing both the training and validation sets) and the inference time required to generate a single prediction. For API-based LLM models (*e.g.,*

[5]https://github.com/cure-lab/LTSF-Linear
[6]https://github.com/wanghq21/MICN
[7]https://github.com/yuqinie98/PatchTST
[8]https://github.com/thuml/Time-Series-Library
[9]https://api.openai.com/v1/
[10]https://huggingface.co/meta-llama

GPT-3.5, GPT-4 and GPT-4o), the total inference time includes the duration required to query the API five times and compute the median prediction. Since these queries are processed in asynchronous threading, The runtime for executing five queries is not five times the runtime of a single query. Moreover, reducing the number of queries to a single run results in significantly degraded performance, making this trade-off necessary. For Local-deployed LLM models (*e.g.,* LLaMA-2-70B, LLaMA-3.1-70B), we measured the time taken to perform a batch of five inference queries. Due to the 40GB GPU memory limitation, the 70B model was loaded in 4-bit mode for inference. All local models, except for the API model, were executed in the same computing environment with an A100-40GB GPU.

B.4 Details of Noise Experiments

Noise Types Cheng et al. (2024) categorize three primary types of noise in time series forecasting: Constant, Missing and Gaussian. Constant noise introduce a fixed deviation from the ground-truth value. These noise can be mathematically represented as:

$$z_A = z + \epsilon, \tag{2}$$

where z_A is the observed value with anomaly, z is the true value, and ϵ is a constant perturbation. Missing noise correspond to completely missing values in the dataset. These noise are defined as:

$$z_A = \epsilon, \tag{3}$$

where ϵ is a constant, often set to zero. Gaussian noise represent deviations that follow a Gaussian distribution. These noise are defined as:

$$z_A = z + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{4}$$

where ϵ is drawn from a normal distribution with mean 0 and variance σ^2. We conducted experiments by injecting three types of noise into the Monash dataset to evaluate the robustness of the LLM to each type of noise.

The single-shot linear models exhibit greater robustness to noise compared to LLM-based models due to the statistical properties of their loss functions. The L1 loss function maintains a constant sum of losses under clean and noisy conditions, ensuring that optimization remains unaffected by noise. Theoretical results (Cheng et al., 2024) support our empirical findings. Since LLMs are token-based models and utilized as zero-shot forecasters, achieving noise robustness is more challenging for them.

Beyond the three representative types of noise discussed above, time-series data often involve more complex challenges such as changes in trend and variations in frequency (Park et al., 2023; Zhou and Yu, 2025). To address this, we additionally conducted experiments comparing LLM-based models and domain-specific models under scenarios involving frequency changes as shown in Table 9.

Noise Filtering Methods Gaussian filtering and EMA filtering are both widely used techniques for noise reduction in signal and image processing. Gaussian filtering applies a Gaussian kernel to smooth data by averaging neighboring values with weights determined by a normal distribution, effectively reducing high-frequency noise while preserving important structures. It is commonly used in image processing to blur and remove noise. On the other hand, EMA filtering is a recursive technique that assigns exponentially decreasing weights to past observations, allowing recent data points to have more influence. This makes EMA particularly effective for real-time applications, such as financial analysis and sensor data smoothing, where responsiveness to recent changes is crucial while still maintaining noise reduction.

However, Gaussian filtering can overly smooth time series data, potentially removing important short-term variations and trends. EMA filtering, while responsive to recent changes, may lag behind true values and be sensitive to sudden spikes. Both methods struggle with non-stationary data, where patterns and noise characteristics change over time. Therefore, applying such noise filtering to a time series before inference may degrade the original dynamics and may not necessarily improve performance.

C Additional Results

C.1 Comparison Between Prompt Methods

Based on the findings of Wei et al. (2022) and Kojima et al. (2022), Chain-of-Thought (CoT) prompting plays a crucial role in enhancing the reasoning capabilities of large language models. Wei et al. (2022) demonstrated that explicitly structuring the reasoning process in prompts allows models to perform complex, multi-step reasoning more effectively. Meanwhile, Kojima et al. (2022) showed that even simple prompts like "Let's think step by step" can significantly improve zero-shot reasoning. Building on these insights, when designing a

Prompt	Input Length	ETTm2		Exchange Rate		Weather	
		MAE	MSE	MAE	MSE	MAE	MSE
LLMTime (b of Figure 4)	384	**0.2322**	**0.1237**	0.3595	0.2893	**0.1880**	**0.1147**
	576	0.2741	0.2037	0.3323	0.2539	0.1943	0.1231
	768	0.3298	0.2539	0.3807	0.4030	0.2075	0.1398
LLMP (b of Figure 5)	384	0.2860	0.1969	0.3436	0.2869	0.2039	0.1200
	576	0.3060	0.2188	0.3336	0.2169	0.1903	0.1167
	768	0.2471	0.1388	0.4171	0.3855	0.1914	0.1197
TS-CoT (a of Figure 6)	384	0.2823	0.1799	0.3647	0.2358	0.1950	0.1451
	576	0.3732	0.3355	0.3297	0.2227	0.2222	0.1310
	768	0.2941	0.2075	0.4083	0.3284	0.1831	0.1666
TS-InContext (b of Figure 6)	384 (1-Shot)	0.2933	0.1909	0.3296	0.2111	0.1921	0.1283
	576 (2-Shot)	0.2678	0.1830	0.3288	0.2044	0.2212	0.2227
	768 (3-Shot)	0.2557	0.1381	**0.3225**	**0.2030**	0.2364	0.3073

Table 5: We compared two recently proposed prompt methods (LLMTime and LLMP) for zero-shot forecasting and two additional prompts (TS-CoT and TS-InContext) using LLM prompting techniques (Chain-of-Thought and In-Context learning), specifically designed for time-series forecasting. LLMTime demonstrates superior performance compared to other prompts with minimal input. In this experiment, we use GPT-4o as a LLM backbone.

prompt for time-series forecasting, it is essential to incorporate CoT principles. This involves guiding the model through a structured, step-by-step approach to analyzing trends, identifying patterns, and making predictions based on historical data. By explicitly prompting the model to reason through the forecasting process, we design TS-CoT Prompt (Figure 6) to improve the accuracy and reliability of predictions in time-series forecasting.

Building on the insights from Brown et al. (2020), In-Context Learning (ICL) serves as a powerful mechanism for improving downstream tasks when using LLMs (Zelikman et al., 2022; Peng et al., 2024). These studies introduced the concept of few-shot learning, demonstrating that language models can generalize from a limited number of examples without fine-tuning. Zelikman et al. (2022) extended this idea with STaR, enabling self-improvement through iterative reasoning, while Peng et al. (2024) proposed ReGenesis, a framework for continual self-improvement via model-generated feedback. By leveraging these principles, we design TS-InContex prompt (Figure 6), which provides a structured set of in-output sequences within the prompt itself, allowing the model to recognize patterns.

C.2 Model Analysis

C.2.1 LLMs' Hyperparameter Analysis

Temperature. We adopt the temperature values reported by LLMTime, which were selected to optimize zero-shot forecasting performance. To assess the sensitivity of LLM performance to this hyperparameter, we conduct an ablation study varying

(a) LLMTime for base models

-12, -13, -15, -7, -11, -6, 43, 98, 43, -10, -11, -9, -11, -12, -9, -10, -12, -8, -9, -13,

Response: -9, -10, -12, -8, -9

(b) LLMTime for instruction-tuned models

System Prompt
You are a helpful assistant that performs time series predictions. The user will provide a sequence and you will predict the remaining sequence. The sequence is represented by decimal strings separated by commas.

User Prompt
Please predict next sequence following input sequence without producing any additional text. Do not say anything like 'the next terms in the sequence are', just return the numbers. Input Sequence: -12, -13, -15, -7, -11, -6, 43, 98, 43, -10, -11, -9, -11, -12, -9, -10, -12, -8, -9, -13,

Response: -9, -10, -12, -8, -9

Figure 4: Two prompts proposed by LLMTime for (a) base LLM models (LLaMA2-7B, 70B, and LLaMA3.1-70B) and (b) instruction-tuned LLM models (GPT-3.5, GPT-4, GPT-4o, and LLaMA-Intstruct).

Models	Temp.	ETTm2		Exchange Rate		Weather	
		MAE	MSE	MAE	MSE	MAE	MSE
GPT-4o	0.1	0.5255	0.9171	0.4823	0.4568	0.2356	0.1662
GPT-4o	0.5	0.3057	0.2383	0.3487	0.2631	0.2463	0.1988
GPT-4o	1.0	0.2322	0.1237	0.3595	0.2893	0.1880	0.1147
DLinear-S	–	0.2039	0.0782	**0.2296**	**0.1166**	0.1643	0.0522
RLinear-S	–	**0.1824**	**0.0677**	0.2571	0.1370	**0.1392**	**0.0469**

Table 6: Forecasting performance by temperatures

the temperature during inference. Table 6 presents the MAE and MSE scores across three datasets.

On ETTm2 and Weather, the default temperature (1.0) yields the best performance. For Exchange Rate, a lower temperature (0.5) marginally improves the error metrics but still fails to outperform linear baselines. These results indicate that while temperature affects performance to a limited degree, the general trend persists: LLMs consistently underperform compared to domain-specific models across all datasets.

Top-p Sampling. Top-p (nucleus) sampling is a widely adopted decoding strategy in modern LLMs, striking a balance between coherence and diversity (Holtzman et al., 2019; Nguyen et al., 2024). It is the default in models such as GPT-4o and LLaMA, and is also used in LLMTime. We perform an ablation study to evaluate how varying the Top-p parameter impacts zero-shot forecasting.

As shown in Table 7, the default value (0.8) leads to strong performance among the tested configurations. However, consistent with the temperature study, GPT-4o remains inferior to linear models

(a) LLMP-Independent (Multi-turn)

System Prompt

User Prompt
0,-12\n 1,-13\n 2,-15\n 3,-7\n 4,-11\n 5,-6\n 6,43\n 7,98\n 8,43\n 9,-10\n 10,-11\n 11,-9\n 12,-11\n 13,-12\n 14,-9\n 15,-10\n 16,-12\n 17,-8\n 18,-9\n 19,-13\n 20,

Response: -9

User Prompt
0,-12\n 1,-13\n 2,-15\n 3,-7\n 4,-11\n 5,-6\n 6,43\n 7,98\n 8,43\n 9,-10\n 10,-11\n 11,-9\n 12,-11\n 13,-12\n 14,-9\n 15,-10\n 16,-12\n 17,-8\n 18,-9\n 19,-13\n 21,

Response: -10

User Prompt
0,-12\n 1,-13\n 2,-15\n 3,-7\n 4,-11\n 5,-6\n 6,43\n 7,98\n 8,43\n 9,-10\n 10,-11\n 11,-9\n 12,-11\n 13,-12\n 14,-9\n 15,-10\n 16,-12\n 17,-8\n 18,-9\n 19,-13\n 22,

Response: -12

User Prompt
0,-12\n 1,-13\n 2,-15\n 3,-7\n 4,-11\n 5,-6\n 6,43\n 7,98\n 8,43\n 9,-10\n 10,-11\n 11,-9\n 12,-11\n 13,-12\n 14,-9\n 15,-10\n 16,-12\n 17,-8\n 18,-9\n 19,-13\n 23,

Response: -8

User Prompt
0,-12\n 1,-13\n 2,-15\n 3,-7\n 4,-11\n 5,-6\n 6,43\n 7,98\n 8,43\n 9,-10\n 10,-11\n 11,-9\n 12,-11\n 13,-12\n 14,-9\n 15,-10\n 16,-12\n 17,-8\n 18,-9\n 19,-13\n 24,

Response: -9

(b) LLMP-Independent (Single-turn)

System Prompt
You are a helpful assistant that performs time series predictions. The user will provide you with a sequence of ordered pairs (x, y), and you will predict y for pairs where only x is given. Each pair is separated by a newline.

User Prompt
Please predict the missing values in the y column based on the given x and y data points without producing any additional text. Do not say anything like 'the next terms in the sequence are', just return only the y values as numbers without x values. x, y\n 0,-12\n 1,-13\n 2,-15\n 3,-7\n 4,-11\n 5,-6\n 6,43\n 7,98\n 8,43\n 9,-10\n 10,-11\n 11,-9\n 12,-11\n 13,-12\n 14,-9\n 15,-10\n 16,-12\n 17,-8\n 18,-9\n 19,-13\n 20, \n 21, \n 22, \n 23, \n 24, \n

Response: -9, -10, -12, -8, -9

Figure 5: (a) is the original prompt proposed in LLMP. The (a) prompt requires independently processing O queries to obtain O output values. This significantly increases the computational cost of using LLMs. To address this issue, we designed a single-turn prompt (b) that enables LLMP to be performed more efficiently.

(a) TS-CoT

System Prompt
You are a helpful assistant that performs time series predictions. The user will provide a sequence and you will predict the remaining sequence. The sequence is represented by decimal strings separated by commas.

User Prompt
Sequence during the input period: -12, -13, -15, -7, -11, -6, 43, 98, 43, -10, -11, -9, -11, -12, -9, -10, -12, -8, -9, -13,
Let's think step by step.

Step 1) Describe the solution process to make future predictions that reflect the description in up to five sentences.

Step 2) Considering the answers to previous steps, please predict next sequence following input sequence without producing any additional text. Do not say anything like 'the next terms in the sequence are', just return the numbers. Input Sequence:-12, -13, -15, -7, -11, -6, 43, 98, 43, -10, -11, -9, -11, -12, -9, -10, -12, -8, -9, -13,

Response:
Answer 1) The sequence provided appears to be non-linear and may contain multiple patterns or regimes. To predict future data points, it's important to identify any apparent periodicity, seasonality, trends, or cycles within the data. The sequence might show different behaviors at different segments, which can include rising trends, falling trends, and various plateaus or periods of relative stability ...
Answer 2) -9, -10, -12, -8, -9

(b) TS-InContext (3-Shot)

System Prompt
You are a helpful assistant that performs time series predictions. The user will provide a sequence and you will predict the remaining sequence. The sequence is represented by decimal strings separated by commas.

User Prompt
We give you input and output sequence samples:
1. Sequence: -12, -13, -15, -7, -11, <sep> -6, 43, 98, 43, -10
2. Sequence: -6, 43, 98, 43, -10, <sep> -11, -9, -11, -12, -9
3. Sequence: -11, -9, -11, -12, -9, <sep> -10, -12, -8, -9, -13

Please predict next sequence following input sequence without producing any additional text. Do not say anything like 'the next terms in the sequence are', just return the numbers. Input Sequence: -10, -12, -8, -9, -13, <sep>

Response: -9, -10, -12, -8, -9

Figure 6: (a) TS-CoT uses a step-by-step reasoning approach (CoT) to guide predictions by analyzing patterns before forecasting. (b) TS-InContext (3-Shot) leverages in-context learning by providing examples of input-output pairs, allowing the model to infer patterns directly from demonstrations.

across all datasets, suggesting that decoding hyperparameters do not fundamentally alter the performance gap.

C.2.2 Evaluating Entire Horizon on Three Real-world Datasets

We adopt the last-sample evaluation protocol introduced by prior work (LLMTime), which evaluates forecasting models only on the final prediction window within each test sequence. This approach substantially reduces the computational and financial costs of repeated LLM inference, particularly when relying on commercial APIs. It is important to clarify that the term last sample does not refer to a single data point, but rather to the final forecasting window for each multivariate time series in the test split. Since the LLM-based forecasters operate independently on each time series, the total number of forecast instances evaluated under this setting is 1,219, as detailed in Appendix Table 3.

To provide a more complete evaluation of temporal generalization, we further conduct experiments using a sliding-window evaluation protocol. In this

Models	Top-p	ETTm2		Exchange Rate		Weather	
		MAE	MSE	MAE	MSE	MAE	MSE
GPT-4o	0.1	0.2816	0.2032	0.3697	0.2930	0.1902	0.1170
GPT-4o	0.5	0.3153	0.2424	0.3913	0.3078	0.2019	0.1235
GPT-4o	0.8	0.2322	0.1237	0.3595	0.2893	0.1880	0.1147
GPT-4o	1.0	0.2825	0.1996	0.3562	0.2926	0.1897	0.1268
DLinear-S	–	0.2039	0.0782	**0.2296**	**0.1166**	0.1643	0.0522
RLinear-S	–	**0.1824**	**0.0677**	0.2571	0.1370	**0.1392**	**0.0469**

Table 7: Forecasting performance by top-p values

Models	ETTm2		Exchange Rate		Weather	
	MAE	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
GPT-4o	0.3391	0.2156	0.3292	0.1824	0.0350	0.0024
LLaMA3.1-8B-Inst.	0.3684	0.2508	0.3619	0.2682	0.0365	0.0026
DLinear-S	0.1455	0.0425	**0.2728**	**0.1105**	0.0242	**0.0010**
RLinear-S	**0.1385**	**0.0411**	0.2974	0.1268	**0.0235**	**0.0010**

Table 8: Forecasting performance on three real-world datasets over the entire test set range.

setup, forecasts are generated across all time steps in the test split by sliding the input window with a stride equal to the prediction length. For consistency and interpretability, we focus our analysis on the main target variable of each dataset (*e.g.,* the oil temperature variable in ETTm2).

As shown in Table 8, the results from this extended evaluation confirm our earlier findings: while LLM-based forecasters such as GPT-4o and LLaMA3.1-8B-Instruct exhibit competitive short-term performance, linear models (*e.g.,* DLinear-S, RLinear-S) continue to outperform them in terms of robustness and accuracy across the entire test horizon. These observations reinforce the limitations of current zero-shot LLM forecasters and underline the temporal stability offered by lightweight, domain-specific architectures.

C.2.3 Evaluating Robustness Against Frequency-Based Noise

Models	Clean		Freq-Replace		Freq-Add	
	MAE	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
GPT-4o	**0.6726**	**0.9089**	0.9065	1.5183	0.7862	1.2615
GPT-4	0.6956	0.9533	0.9171	1.4361	0.7693	1.0061
DLinear-S	0.7074	0.8890	**0.7059**	0.8899	**0.7081**	0.8902
RLinear-S	0.7182	0.8804	0.7129	**0.8733**	0.7183	**0.8743**

Table 9: Forecasting results of each model on the Monash dataset with two periodic noises.

To further investigate the robustness of forecasting models under structured perturbations, we conduct additional experiments using periodic noise. Two variants of noise are considered: (1) Freq-Add, in which a sinusoidal component is added to the original time series, and (2) Freq-Replace, where the original series is replaced entirely by a sinusoidal signal. These settings are designed to emulate scenarios where time-series data are affected by periodic interference unrelated to the true signal.

Experimental results in Table 9 show that LLM-based models, such as GPT-4 and GPT-4o, experience substantial performance degradation in the presence of such periodic distortions. In contrast, linear baselines such as DLinear-S and RLinear-S remain largely unaffected, demonstrating strong resilience to this form of noise. This observation aligns with previous findings, reinforcing that linear models exhibit superior robustness when exposed to structured, non-informative patterns in the input.

C.2.4 Comparison with Additional Baselines

Models	ETTm2		Exchange Rate		Weather	
	MAE	MSE	MAE	MSE	MAE	MSE
GPT-4o	0.2322	0.1237	0.3595	0.2893	0.1880	0.1147
ARIMA	0.3402	0.2201	0.3460	0.2535	0.3104	0.2722
N-BEATS	0.2804	0.1842	0.3808	0.3079	0.2169	0.1209
DLinear-S	0.2039	0.0782	**0.2296**	**0.1166**	0.1643	0.0522
RLinear-S	**0.1824**	**0.0677**	0.2571	0.1370	**0.1392**	**0.0469**

Table 10: Forecasting results of ARIMA and N-BEATS on three real-world datasets

Statistical and lightweight forecasting models, such as ARIMA (Box and Jenkins, 1968) and N-BEATS (Oreshkin et al., 2019), are widely recognized for their simplicity and computational efficiency. To ensure a comprehensive evaluation, we included these models in our benchmark and report their forecasting performance on three real-world datasets in Table 10. The results show that, while LLM-based zero-shot models (*e.g.,* GPT-4o) can outperform ARIMA and N-BEATS in certain settings, their performance is still surpassed by simpler linear models such as DLinear-S and RLinear-S trained on a single input sequence. These single-shot linear models not only achieve higher forecasting accuracy but also retain strong computational efficiency. We propose these models as new, competitive baselines for zero-shot forecasting with LLMs, providing a more challenging yet realistic reference point for future research.

C.3 Qualitative Results of Forecasting Models

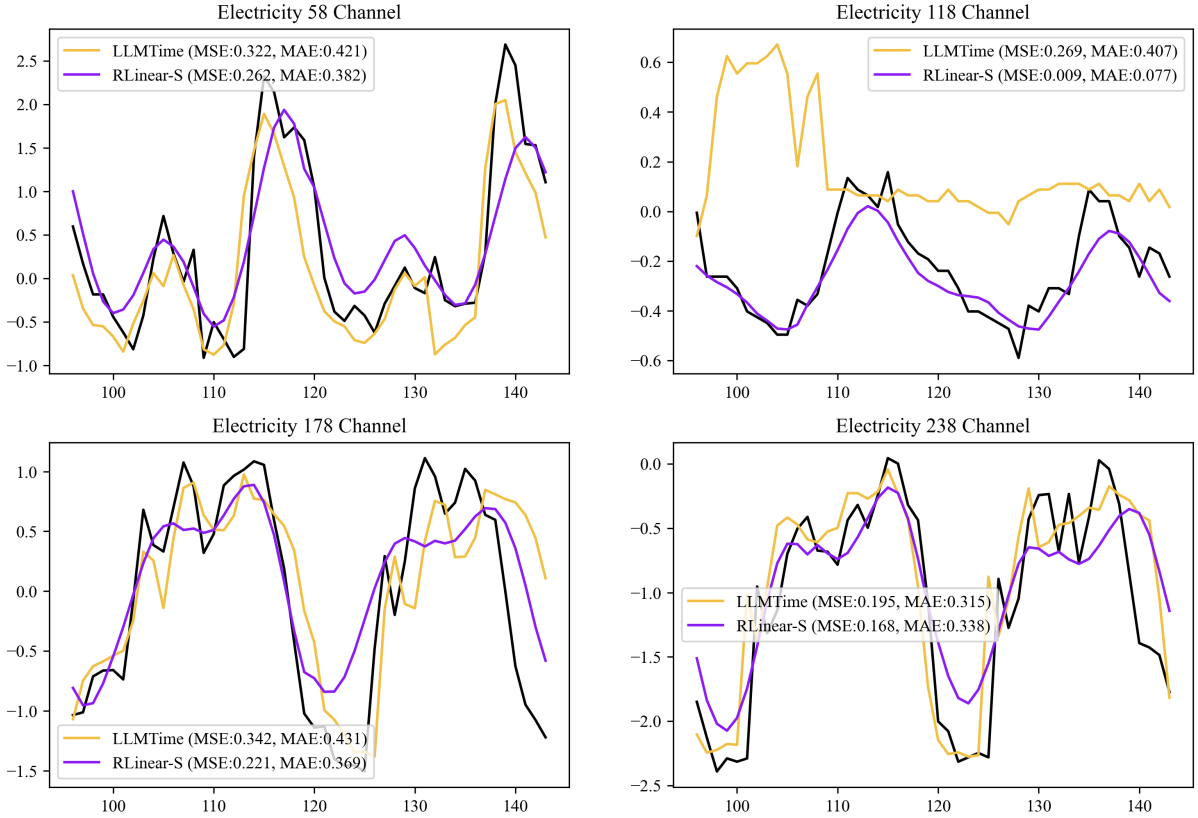


Figure 7: Qualitative results on Electricity dataset across four channels (58, 118, 178, and 238). The black line represents ground truth, while LLMTime (yellow) and RLinear-S (purple) show model predictions with corresponding MSE and MAE values. Overall, RLinear-S tends to outperform LLMTime.

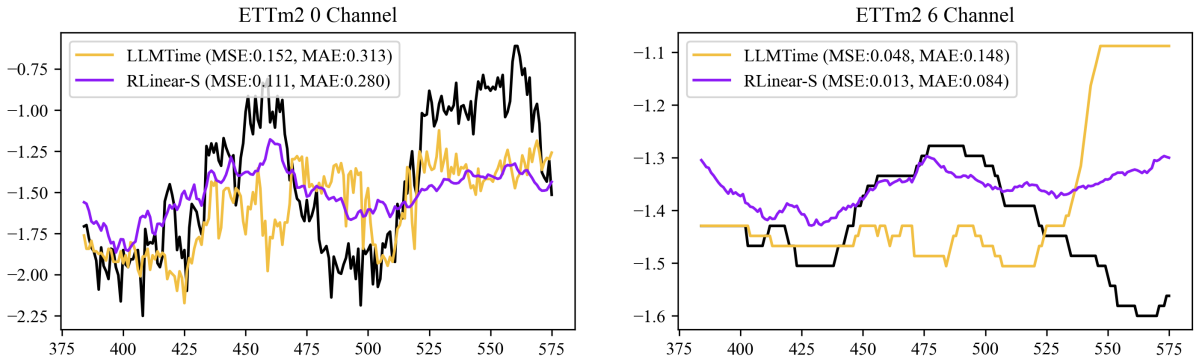


Figure 8: Qualitative results on ETTm2 dataset across four channels (0 and 6

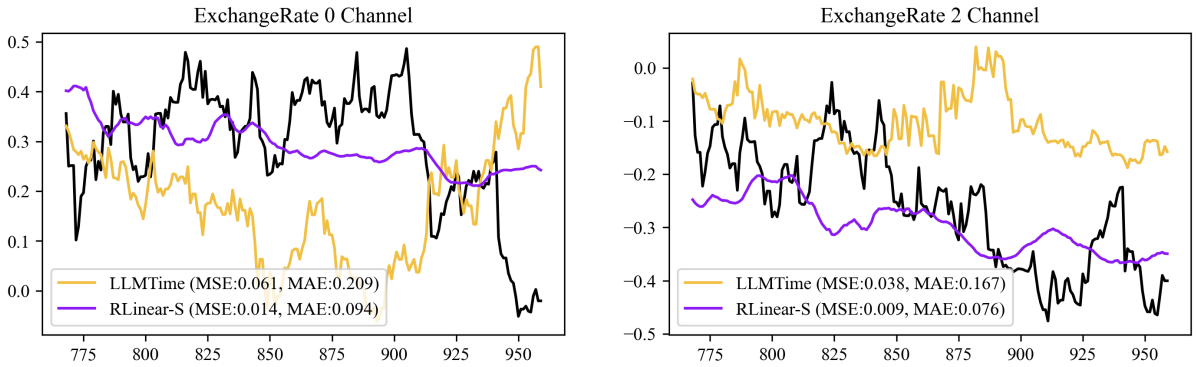


Figure 9: Qualitative results on ExchangeRate dataset across four channels (0 and 2)

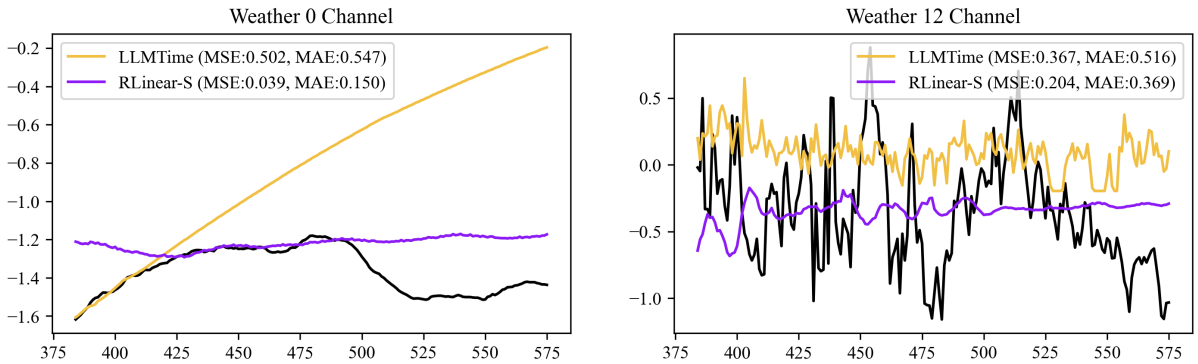


Figure 10: Qualitative results on Weather dataset across four channels (0 and 12)

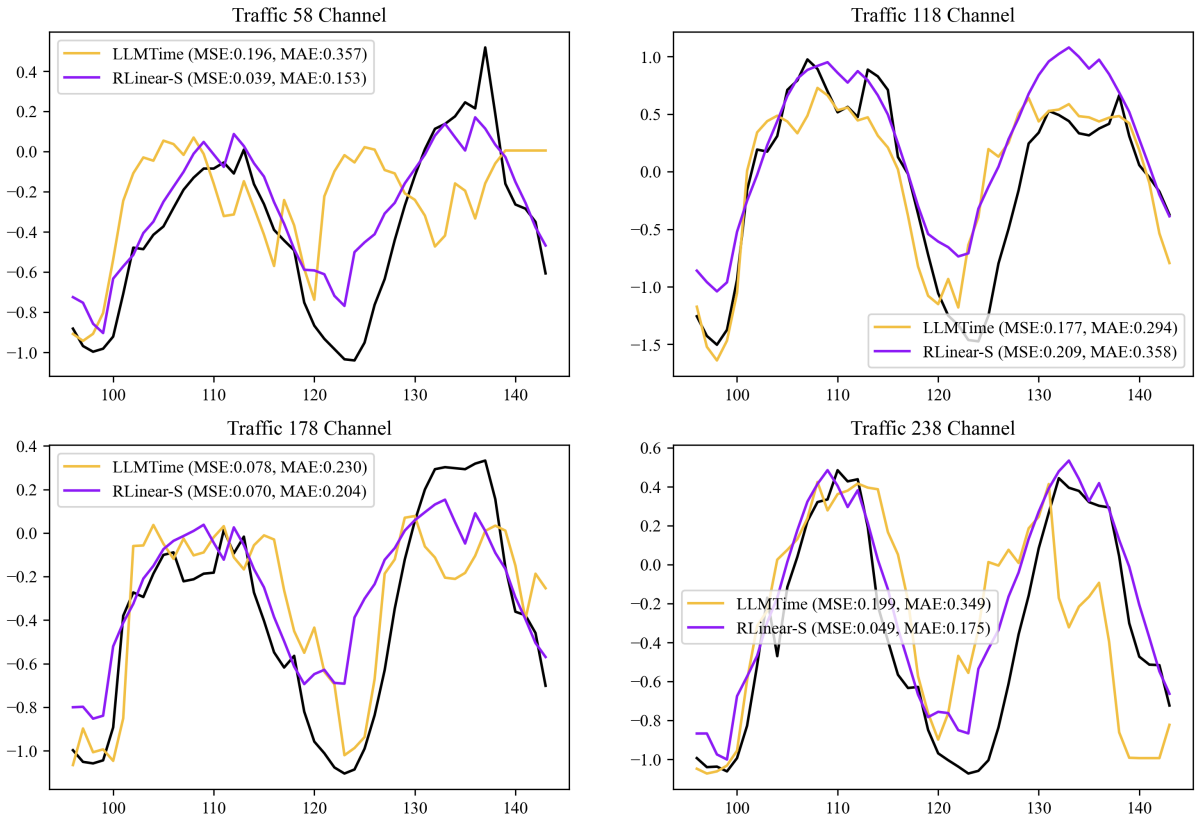


Figure 11: Qualitative results on Traffic dataset across four channels (58, 118, 178, and 238).

C.4 Full Results

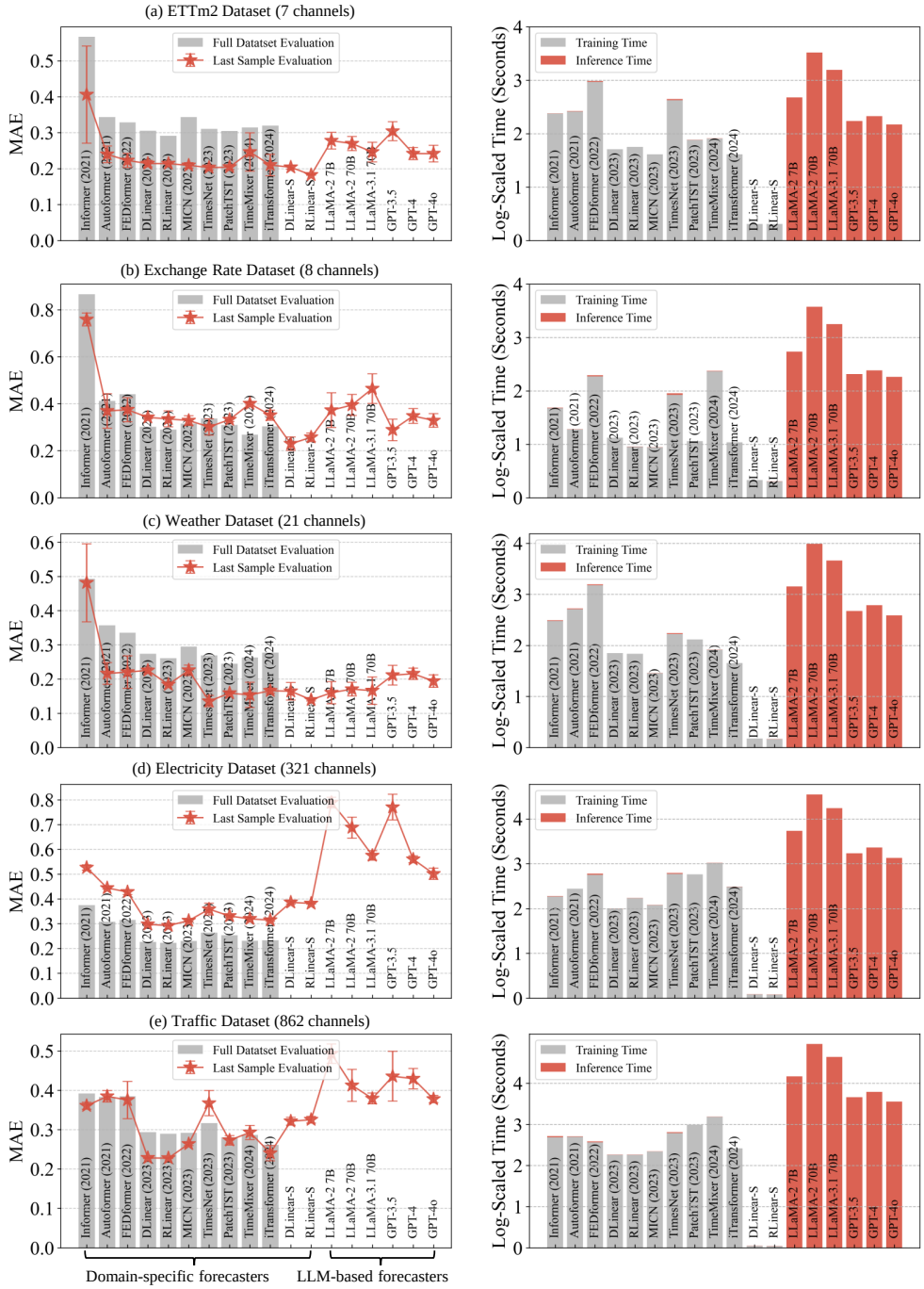


Figure 12: Multivariate forecasting results on five datasets from the Informer benchmark. We report the normalized MAE and log-scaled inference time for domain-specific and LLM-based forecasters with confidence intervals.

Transferring Textual Preferences to Vision-Language Understanding through Model Merging

Chen-An Li **Tzu-Han Lin** **Yun-Nung Chen** **Hung-yi Lee**
National Taiwan University, Taipei, Taiwan
{r13942069,r12944034}@ntu.edu.tw y.v.chen@ieee.org hungyilee@ntu.edu.tw

Abstract

Large vision-language models (LVLMs) perform outstandingly across various multimodal tasks. However, their ability to evaluate generated content remains limited, and training vision-language reward models (VLRMs) with preference data is computationally expensive. This paper explores a training-free alternative by merging text-based reward models (RMs) with LVLMs to create VLRMs. Our approach shows that integrating these models leads to improved performance over LVLMs' scoring and text-based RMs, offering an efficient method for incorporating textual preferences into LVLMs. The code and data are publicly available at https://github.com/lca0503/MergeToVLRM.

1 Introduction

Large vision-language models (LVLMs) have shown exceptional performance across a wide range of multimodal tasks (Hurst et al., 2024; Team et al., 2024; Anthropic, 2024), primarily due to the implementation of reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022), which utilizes preference data (Sun et al., 2024; Li et al., 2024b). This process often requires the use of reward models (RMs). However, LVLMs still struggle to assess generated content effectively (Chen et al., 2024a; Li et al., 2024a), and training an RM with preference data is resource-intensive.

In this work, we investigate an alternative approach: *Can knowledge derived from text-only preference data be transferred to LVLMs without additional training?* Several state-of-the-art LVLMs are built upon pre-trained language models with vision encoders and adapters (Dubey et al., 2024; Team, 2025; Lu et al., 2024). This architectural design suggests that textual preferences learned by text-based RMs may potentially integrate into LVLMs through parameter merging.

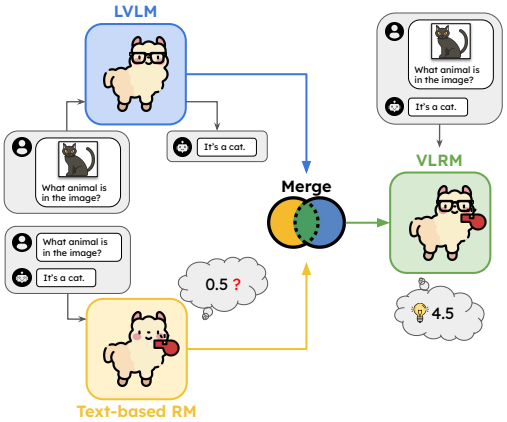


Figure 1: Framework for merging a text-based RM with an LVLM. LVLMs excel at visual tasks, while text-based RMs struggle to provide accurate rewards without visual cues. We transfer textual preferences to the vision-language understanding, resulting in a VLRM. All icons used in this figure are sourced from https://www.flaticon.com/

Building on this idea, we propose merging LVLMs with text-based RMs to create vision-language reward models (VLRMs), as illustrated in Figure 1. Our approach leverages existing RMs and LVLMs, eliminating the need for costly multimodal preference data collection and training. We explore various merging strategies, ranging from simple weighted averaging (Wortsman et al., 2022) to advanced techniques such as task arithmetic (Ilharco et al., 2023), TIES (Yadav et al., 2024), and DARE (Yu et al., 2024a).

We assess performance using VL-RewardBench (Li et al., 2024a) and Best-of-N sampling with TextVQA (Singh et al., 2019) and MMMU-Pro (Yue et al., 2024b). The results show that our combined VLRMs outperform scoring through LVLMs and reward generation with text-based RMs. Our approach offers a training-free method for transferring textual preferences to LVLMs via model merging, and we provide a detailed analysis of merging strategies, demonstrating its effectiveness across multiple benchmarks.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 923–943
July 27 - August 1, 2025

2 Related Work

Preference Dataset A common approach to train a reward model is to use the Bradley–Terry model (Bradley and Terry, 1952), which relies on paired data for learning. In NLP, many high-quality preference datasets are already available (Stiennon et al., 2020; Bai et al., 2022; Ethayarajh et al., 2022; Köpf et al., 2023; Cui et al., 2024; Zhu et al., 2024; Wang et al., 2024). Similarly, in the vision-language domain, several preference datasets have been introduced (Yu et al., 2024b,c; Chen et al., 2024b; Wijaya et al., 2024; Li et al., 2024c; Zhou et al., 2024; Xiao et al., 2024). In this work, we explore the potential of transferring textual preferences to LVLMs in a training-free manner, specifically through model merging.

LVLM-as-a-Judge & Evaluation LVLM-as-a-Judge refers to utilizing strong large vision-language models for evaluation and judgment. These LVLMs can be either closed-source (OpenAI, 2023; Hurst et al., 2024; Team et al., 2024; Anthropic, 2024) or open-source (Lee et al., 2024; Dubey et al., 2024; Deitke et al., 2024; Team, 2025). To assess LVLMs as generative reward models, Chen et al. (2024a) established benchmarks and found that LVLMs exhibit high agreement with humans in pairwise comparison judgments, but perform poorly in scoring evaluation and batch ranking tasks. Recently, VL-RewardBench (Li et al., 2024a) introduced challenging cases and complex multimodal reasoning tasks, revealing that most off-the-shelf LVLMs struggle with such evaluations.

Model Merging Model merging is a common, training-free method for combining skills from multiple models within the parameter space. A basic approach involves simple weighted averaging (Wortsman et al., 2022), while more advanced techniques have been developed (Yadav et al., 2024; Yu et al., 2024a; Yang et al., 2024). These techniques have already proven effective in reward modeling (Rame et al., 2024; Lin et al., 2024) and LLM-as-a-judge (Kim et al., 2024) in NLP. Recently, REMEDY (Zhu et al., 2025) introduced strategies for merging LVLMs. In contrast, our work focuses on merging textual reward models into the language modeling components of LVLMs.

3 Methodology

We propose a training-free method to transfer textual preferences from a text-based RM θ^{RM} to a LVLM θ^{LVLM} through model merging.

Since both models originate from the same pre-trained language model θ^{PRE}, we merge modules that appear in both models and preserve the LVLM's vision capabilities and text-based RM reward function, resulting in a VLRM that can assess textual and visual content without additional training. Below, we outline the components and merging strategies involved.

3.1 Model Components

The pre-trained language model consists of:

$$\theta^{\text{PRE}} = \{\theta^{\text{PRE}}_{\text{emb}}, \theta^{\text{PRE}}_{\text{trans}}, \theta^{\text{PRE}}_{\text{lm}}\},$$

where $\theta^{\text{PRE}}_{\text{emb}}$ is the embedding layer, $\theta^{\text{PRE}}_{\text{trans}}$ is the transformer, and $\theta^{\text{PRE}}_{\text{lm}}$ is the language modeling head, which maps the final hidden state of the transformer to the vocabulary.

The LVLM expands upon this with:

$$\theta^{\text{LVLM}} = \{\theta^{\text{LVLM}}_{\text{venc}}, \theta^{\text{LVLM}}_{\text{adapt}}, \theta^{\text{LVLM}}_{\text{emb}}, \theta^{\text{LVLM}}_{\text{trans}}, \theta^{\text{LVLM}}_{\text{lm}}\},$$

where $\theta^{\text{LVLM}}_{\text{venc}}$ is the vision encoder, and $\theta^{\text{LVLM}}_{\text{adapt}}$ is the adapter that integrates the vision encoder outputs into the language model.

Similarly, the text-based RM is defined as:

$$\theta^{\text{RM}} = \{\theta^{\text{RM}}_{\text{emb}}, \theta^{\text{RM}}_{\text{trans}}, \theta^{\text{RM}}_{\text{rm}}\},$$

where $\theta^{\text{RM}}_{\text{rm}}$ is the reward modeling head, which projects the transformer's final hidden state to a scalar value as the reward for a given input.

3.2 Merging Strategies

We explore four merging strategies.

Weighted Averaging The weighted averaging strategy is defined as:

$$\theta^{\text{MERGE}}_{\text{trans}} = \lambda \cdot \theta^{\text{LVLM}}_{\text{trans}} + (1-\lambda) \cdot \theta^{\text{RM}}_{\text{trans}},$$

where λ is a hyperparameter that controls the weight distribution between the two terms.

Task Arithmetic Task arithmetic strategy is defined as:

$$\begin{aligned}
\tau^{\text{LVLM}} &= \theta^{\text{LVLM}}_{\text{trans}} - \theta^{\text{PRE}}_{\text{trans}}, \\
\tau^{\text{RM}} &= \theta^{\text{RM}}_{\text{trans}} - \theta^{\text{PRE}}_{\text{trans}}, \\
\theta^{\text{MERGE}}_{\text{trans}} &= \theta^{\text{PRE}}_{\text{trans}} + \lambda \cdot \tau_{\text{LVLM}} + \lambda \cdot \tau_{\text{RM}},
\end{aligned}$$

where τ^{LVLM} represents the task vector derived from instruction tuning, and τ^{RM} is the task vector obtained from reward modeling. The hyperparameter λ controls the contribution of the task vectors.

Method	VL-RewardBench					TextVQA	MMMU-Pro	
	General	Hallucination	Reasoning	Overall	Macro Avg.	Overall	Standard	Vision
Llama-3.2-Vision	33.3*	38.4*	56.6*	42.9*	42.8*	46.4	28.8	19.8
Tulu-2.5-RM	43.2	31.4	54.1	38.9	42.9	42.6	29.8	21.4
Random	**50.0**	50.0	50.0	50.0	50.0	48.2	29.2	18.4
Cascade	44.8	37.8	57.2	43.8	46.6	43.2	30.9	**23.4**
Linear	39.3	52.3	54.4	51.0	48.7	54.7	27.8	22.1
Task Vec.	48.6	59.4	59.7	57.9	55.9	59.0	31.0	22.7
TIES	43.7	58.2	58.5	56.2	53.5	**64.2**	29.1	22.6
DARE + Task Vec.	49.2	**61.7**	**61.0**	**59.7**	**57.3**	58.8	30.3	22.4
DARE + TIES	49.2	59.1	58.2	57.4	55.5	57.3	**31.6**	22.0

Table 1: Comparison of merging methods across the VL-RewardBench, TextVQA, and MMMU-Pro datasets using TULU-2.5-RM for merging. *Indicates results from Li et al. (2024a).

TIES & DARE For the TIES and DARE strategies, we simplify the expression to:

$$\theta_{\text{trans}}^{\text{MERGE}} = \theta_{\text{trans}}^{\text{PRE}} + \lambda \cdot f(\tau^{\text{LVLM}}, d) + \lambda \cdot f(\tau^{\text{RM}}, d),$$

where $f(\cdot)$ denotes the function for trimming, selecting, and rescaling the task vector, and d is the density determining how many parameters are retained. The two strategies apply different methods for trimming, selecting, and rescaling. See Appendix A for more details on TIES and DARE.

3.3 Merged VLRM

The merged embedding parameters, $\theta_{\text{emb}}^{\text{MERGE}}$ are obtained following standard embedding merging techniques outlined in MergeKit (Goddard et al., 2024), as detailed in Appendix A.

Finally, the merged VLRM θ^{MERGE} is obtained by combining several components:

$$\theta^{\text{MERGE}} = \{\theta_{\text{venc}}^{\text{LVLM}}, \theta_{\text{adapt}}^{\text{LVLM}}, \theta_{\text{emb}}^{\text{MERGE}}, \theta_{\text{trans}}^{\text{MERGE}}, \theta_{\text{rm}}^{\text{RM}}\},$$

As a result, the merged VLRM can be used to provide rewards for both text and image content.

4 Experiments

4.1 Experimental Setup

4.1.1 Models

In this paper, we employ Llama-3.2-11B-Vision-Instruct (Dubey et al., 2024) as our LVLM, referred to as Llama-3.2-Vision. For text-based RMs, we use Llama-3.1-Tulu-2-8B-uf-mean-rm (Ivison et al., 2024) and Llama-3.1-Tulu-3-8B-RM (Lambert et al., 2024), which we denote as Tulu-2.5-RM and Tulu-3-RM, respectively. All models derive from the same pre-trained language model Llama-3.1-8B. Our main results focus on Tulu-2.5-RM since it outperforms Tulu-3-RM on several VQA tasks with text-based input. Please refer to Appendix E for the model details.

4.1.2 Model Merging

We use MergeKit for model merging and apply several techniques: weighted averaging, task arithmetic, TIES, and DARE—labeled as Linear, Task Vec., TIES, and DARE, respectively. Additionally, we explore combining DARE with task arithmetic and TIES for a more thorough analysis. To determine the optimal merging hyperparameters, we conduct a hyperparameter search and sample 400 instances from the RLAIF-V (Yu et al., 2024c) training set as our validation set. More details are provided in Appendix A.

4.2 Reward Model Evaluation

4.2.1 VL-RewardBench

We assess the merged VLRMs using VL-RewardBench (Li et al., 2024a), a benchmark that includes three domains: general multimodal instructions, hallucination-related tasks, and multimodal reasoning tasks. Each instance includes a multimodal query that consists of an image and a user prompt, along with a chosen response and a rejected response.

4.2.2 Best-of-N Sampling

We assess our reward model's effectiveness in enhancing performance through reranking using Best-of-N sampling, where N = 8 in our work. This method scores and ranks responses to check if the highest-scoring one matches the correct answer. Specifically, we use Llama-3.2-11B-Vision-Instruct to generate eight candidates for the TextVQA (Singh et al., 2019) and MMMU-Pro (Yue et al., 2024b) datasets. See Appendix B for dataset details.

4.3 Main Results

Table 1 demonstrates the effectiveness of merging methods for combining an LVLM with

a text-based RM. The baseline approaches include `Llama-3.2-Vision`, which utilizes the LVLM for direct scoring—pairwise scoring in VL-RewardBench and verbalized scoring in Best-of-N sampling tasks. Another baseline method, `Tulu-2.5-RM`, utilizes the text-based RM that focuses solely on evaluating the textual elements of questions and responses. We also incorporate a `Random` baseline that randomly selects responses. Furthermore, we implement a `Cascade` approach that employs a two-stage process: it first uses the LVLM to generate text descriptions of images based on the given question, then passes these descriptions with the original text inputs through the text-based RM to produce final scores.

As shown in Table 1, merged VLRMs consistently outperform `Llama-3.2-Vision` and `Tulu-2.5-RM` across nearly all merging methods and benchmarks. This result demonstrates that combining a text-based RM with an LVLM effectively transfers textual preferences without training. Different merging strategies achieve the highest scores in different benchmarks, but overall, more advanced methods outperform simpler ones, highlighting the advantages of structured merging techniques. Additionally, in several benchmarks, merged VLRMs surpass or match the strong `Cascade` baseline, suggesting that model merging captures more information than merely cascading two models. Furthermore, as shown in Table 2, our merged VLRMs even exceed the performance of the 90B LVLM and achieve results comparable to commercial models. A similar trend emerges when using `Tulu-3-RM` as the text-based RM; further details are provided in Appendix G.1.

Method	General	Hallucination	Reasoning
*Open-Source Models**			
`Llama-3.2-Vision (11B)`	33.3	38.4	56.6
`Llama-3.2-Vision (90B)`	42.6	57.3	61.7
*Proprietary Models**			
`Gemini-1.5-Flash`	47.8	59.6	58.4
`Gemini-1.5-Pro`	50.8	72.5	64.2
`GPT-4o-mini`	41.7	34.5	58.2
`GPT-4o`	49.1	67.6	70.5
Using TULU-2.5-RM for merging			
`Linear`	39.3	52.3	54.4
`Task Vec.`	48.6	59.4	59.7
`TIES`	43.7	58.2	58.5
`DARE + Task Vec.`	49.2	61.7	61.0
`DARE + TIES`	49.2	59.1	58.2

Table 2: VL-RewardBench results comparing open-source and proprietary models with our reward model using TULU-2.5-RM for merging. *Indicates results from Li et al. (2024a). Full results are provided in Table 12

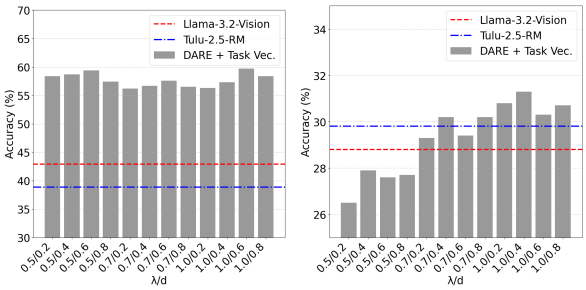
	VL-RB	**TextVQA**	**MMMU-Pro**	
Method	Overall	Overall	Standard	Vision
`Linear`	51.0	54.7	27.8	22.1
`w/o image input`	39.8	45.8	29.1	21.6
`Task Vec.`	57.9	59.0	31.0	22.7
`w/o image input`	44.9	38.7	31.8	21.0
`TIES`	56.2	64.2	29.1	22.6
`w/o image input`	42.7	40.9	31.2	21.0
`DARE + Task Vec.`	59.7	58.8	30.3	22.4
`w/o image input`	44.5	36.2	32.1	20.8
`DARE + TIES`	57.4	57.3	31.6	22.0
`w/o image input`	45.6	36.9	32.1	20.8

Table 3: Comparison of merging methods with and without image input, using `Tulu-2.5-RM` for merging. VL-RB stands for VL-RewardBench.

4.4 Analysis

Without Image Input To further investigate whether the merged VLRMs effectively use the vision encoder, we conduct an ablation study by evaluating the models without image input. As shown in Table 3, most models with image input outperform those without it across various merging techniques. This result suggests that the vision encoder plays an active role after merging, with performance gains not solely attributed to the text-based RM. These findings highlight how merging methods effectively combine textual and visual information. However, image input does not improve performance in the MMMU-Pro Standard set, likely because this set emphasizes reasoning, where reward assessments depend more on textual coherence than visual understanding, limiting the vision encoder's contribution. A similar trend occurs when using `Tulu-3-RM` as the text-based RM; see Appendix G.2 for details.

Effect of Merging Hyperparameters We also investigate how merging hyperparameters impacts performance. Figure 2 presents the results of searching for d within the range [0.2, 0.4, 0.6, 0.8] and λ within [0.5, 0.7, 1.0] for `DARE + Task Vec.`. Our findings indicate that optimal hyperparameter values vary across benchmarks. For example, in VL-RewardBench, λ values do not have a significant effect, but in the MMMU-Pro standard set, we observe that $\lambda = 1.0$ performs best. This variation indicates that the choice of hyperparameters affects the performance of the final merged VLRM differently across tasks. Consequently, it highlights the



(a) VL-RewardBench (b) MMMU-Pro (Standard)

Figure 2: Effect of `Dare + Task Vec.` merging hyperparameters with `Tulu-2.5-RM` as the text-based RM.

importance of a well-curated validation set when selecting the optimal hyperparameters, which could be further explored in future research.

Furthermore, our results for d align with previous studies on TIES and DARE: even when task vectors are trimmed to lower rates (e.g., 0.4, 0.2), the merged VLRMs maintain strong performance, consistent with the findings on LLM merging. For further hyperparameter search results across other methods and benchmarks, refer to Appendix G.3.

Computation Overhead In our experiments, model merging is done entirely on CPUs (Intel Xeon Silver 4216) using a system with 128 GB of RAM. Using 11 different λ values for weighted averaging takes about 1.5 hours of CPU time. The task arithmetic method takes a similar amount of time when using the same number of λ values. Applying 12 combinations of λ and density d for the TIES method takes about 6 hours of CPU time, while DARE takes around 3 hours to handle the same number of combinations.

We evaluate the models on a validation set of 400 examples from the RLAIF-V dataset. We run model inference on GPUs with 24 GB of memory (Nvidia GeForce RTX 3090). Across all configurations and merging methods, inference takes approximately 1.5 hours of GPU time per method.

Overall, merging and evaluation require much less computing time than training a reward model from scratch. Since merging is the most time-consuming step and runs only on the CPU, the total computational cost stays relatively low. Also, both merging and evaluation can be run in parallel on multiple machines to reduce the actual runtime.

5 Conclusion

This work presents a training-free approach for integrating text-based RMs into LVLMs through model merging. Our method enables the efficient transfer of textual preferences without the expensive multimodal preference data collection or additional training. Experimental results show that our approach outperforms LVLM scoring and text-based RMs in multimodal reward assessment tasks.

Limitations

Our study has several limitations. First, we focused on a specific 11B vision-language model paired with an 8B text-based reward model, primarily due to limitations in computational resources. Additionally, we focused solely on the LLaMA architecture and did not explore alternatives like Qwen (Bai et al., 2023a,b) due to the absence of a suitable Qwen-based reward model for our experiments. Furthermore, we did not perform extensive ablation studies on the validation set. Our experimental results highlight the importance of a well-curated validation set in selecting optimal hyperparameters, which could be explored further in future research. Finally, due to the sensitivity of RLHF to hyperparameter tuning and our computational constraints, we did not implement algorithms like PPO (Schulman et al., 2017). Future work could explore integrating RLHF with merged VLRMs to assess its potential impact.

Ethics Statement

Our approach leverages pre-trained language and reward models, which may inherit biases from the training data. While merging models can enhance efficiency, it does not inherently mitigate existing biases. We encourage further research to evaluate and address potential biases in merged models to ensure fairness across diverse user groups.

Acknowledgements

We thank the reviewers for their insightful comments. This work was financially supported by the National Science and Technology Council (NSTC) in Taiwan, under Grant 113-2628-E-002-033. We thank to National Center for High-performance Computing (NCHC) of National Applied Research Laboratories (NARLabs) in Taiwan for providing computational and storage resources. We are also grateful to Yu-Xiang Lin from National Taiwan University for his valuable advice and thoughtful discussions.

References

Anthropic. 2024. Claude 3.5 sonnet. Accessed: 2025-02-04.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023a. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023b. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. 2024a. MLLM-as-a-judge: Assessing multimodal LLM-as-a-judge with vision-language benchmark. In *Forty-first International Conference on Machine Learning*.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2024b. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pages 370–387. Springer.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. ULTRAFEEDBACK: Boosting language models with scaled AI feedback. In *Forty-first International Conference on Machine Learning*.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. 2024. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee's MergeKit: A toolkit for merging large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 477–485, Miami, Florida, US. Association for Computational Linguistics.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. 2024. Unpacking DPO and PPO: Disentangling best practices for learning from preference feedback. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri,

David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations - democratizing large language model alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 47669–47681. Curran Associates, Inc.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. 2024. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Seongyun Lee, Seungone Kim, Sue Park, Geewook Kim, and Minjoon Seo. 2024. Prometheus-vision: Vision-language model as a judge for fine-grained evaluation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11286–11315, Bangkok, Thailand. Association for Computational Linguistics.

Lei Li, Yuancheng Wei, Zhihui Xie, Xuqing Yang, Yifan Song, Peiyi Wang, Chenxin An, Tianyu Liu, Sujian Li, Bill Yuchen Lin, et al. 2024a. Vlrewardbench: A challenging benchmark for vision-language generative reward models. *arXiv preprint arXiv:2411.17451*.

Lei Li, Zhihui Xie, Mukai Li, Shunian Chen, Peiyi Wang, Liang Chen, Yazheng Yang, Benyou Wang, Lingpeng Kong, and Qi Liu. 2024b. VLFeedback: A large-scale AI feedback dataset for large vision-language models alignment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6227–6246, Miami, Florida, USA. Association for Computational Linguistics.

Lei Li, Zhihui Xie, Mukai Li, Shunian Chen, Peiyi Wang, Liang Chen, Yazheng Yang, Benyou Wang, Lingpeng Kong, and Qi Liu. 2024c. Vlfeedback: A large-scale ai feedback dataset for large vision-language models alignment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6227–6246.

Tzu-Han Lin, Chen-An Li, Hung-yi Lee, and Yun-Nung Chen. 2024. DogeRM: Equipping reward models with domain knowledge through model merging. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15506–15524, Miami, Florida, USA. Association for Computational Linguistics.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. 2024. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*.

OpenAI. 2023. Gpt-4v system card. Accessed: 2025-02-04.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Alexandre Rame, Nino Vieillard, Leonard Hussenot, Robert Dadashi, Geoffrey Cideron, Olivier Bachem, and Johan Ferret. 2024. WARM: On the benefits of weight averaged reward models. In *Forty-first International Conference on Machine Learning*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. 2024. Aligning large multimodal models with factually augmented RLHF. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13088–13110, Bangkok, Thailand. Association for Computational Linguistics.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Qwen Team. 2025. Qwen2.5-vl.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024. Helpsteer 2: Open-source dataset for training top-performing reward models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Robert Wijaya, Ngoc-Bao Nguyen, and Ngai-Man Cheung. 2024. Multimodal preference data synthetic alignment with reward model. *Preprint*, arXiv:2412.17417.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR.

Wenyi Xiao, Ziwei Huang, Leilei Gan, Wanggui He, Haoyuan Li, Zhelun Yu, Hao Jiang, Fei Wu, and Linchao Zhu. 2024. Detecting and mitigating hallucination in large vision language models via fine-grained ai feedback. *arXiv preprint arXiv:2404.14233*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2024. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *Preprint*, arXiv:2408.07666.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024a. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, and Tat-Seng Chua. 2024b. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13807–13816.

Tianyu Yu, Haoye Zhang, Yuan Yao, Yunkai Dang, Da Chen, Xiaoman Lu, Ganqu Cui, Taiwen He, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2024c. Rlaif-v: Aligning mllms through open-source ai feedback for super gpt-4v trustworthiness. *arXiv preprint arXiv:2405.17220*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 2024a. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. 2024b. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2409.02813*.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. 2024. Lmms-eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Yiyang Zhou, Chenhang Cui, Rafael Rafailov, Chelsea Finn, and Huaxiu Yao. 2024. Aligning modalities in vision large language models via preference fine-tuning. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, Karthik Ganesan, Wei-Lin Chiang, Jian Zhang, and Jiantao Jiao. 2024. Starling-7b: Improving helpfulness and harmlessness with RLAIF. In *First Conference on Language Modeling*.

Didi Zhu, Yibing Song, Tao Shen, Ziyu Zhao, Jinluan Yang, Min Zhang, and Chao Wu. 2025. REMEDY: Recipe merging dynamics in large vision-language models. In *The Thirteenth International Conference on Learning Representations*.

A Merging Details

Weighted Averaging Wortsman et al. (2022) showed that combining the weights of multiple models fine-tuned with varying hyperparameter settings often leads to improved accuracy and robustness. In this work, we employ a weighted averaging strategy as a straightforward method to merge a large vision-language model with a text-based reward model. The weighted averaging strategy is formally defined as:

$$\theta_{\text{trans}}^{\text{MERGE}} = \lambda \cdot \theta_{\text{trans}}^{\text{LVLM}} + (1-\lambda) \cdot \theta_{\text{trans}}^{\text{RM}},$$

where λ is a hyperparameter that determines the weight distribution between the two models. We explore λ values in the range: [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0].

Task Arithmetic Ilharco et al. (2023) demonstrated that the task vector, obtained by subtracting the weights of a pre-trained model from those of the same model after fine-tuning for a specific task, defines the task direction. Utilizing this task vector can improve task performance. We also apply the task arithmetic approach to develop a vision-language reward model. The task arithmetic strategy is formally defined as:

$$\tau^{\text{LVLM}} = \theta_{\text{trans}}^{\text{LVLM}} - \theta_{\text{trans}}^{\text{PRE}},$$
$$\tau^{\text{RM}} = \theta_{\text{trans}}^{\text{RM}} - \theta_{\text{trans}}^{\text{PRE}},$$
$$\theta_{\text{trans}}^{\text{MERGE}} = \theta_{\text{trans}}^{\text{PRE}} + \lambda \cdot \tau_{\text{LVLM}} + \lambda \cdot \tau_{\text{RM}},$$

where τ^{LVLM} denotes the task vector derived from instruction tuning, and τ^{RM} refers to the task vector obtained from reward modeling. The hyperparameter λ controls the relative contribution of task vectors. We explore λ values in the range: [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0].

TIES Yadav et al. (2024) consider the interference between parameters from different models during the model merging process. Their approach consists of three main steps. First, they prune task vector values based on magnitude, retaining only a proportion d of the task vector. Second, they resolve sign conflicts by calculating the total magnitude of parameter values in positive and negative directions and selecting the direction with the larger total magnitude. Only values that match the chosen sign are retained. Finally, they compute the mean of the retained values to determine the final parameter value. The TIES method can be simply expressed as:

$$\theta_{\text{trans}}^{\text{MERGE}} = \theta_{\text{trans}}^{\text{PRE}} + \lambda \cdot f(\tau^{\text{LVLM}}, d) + \lambda \cdot f(\tau^{\text{RM}}, d),$$

where $f(\cdot)$ denotes the function for trimming, selecting, and rescaling the task vector, and d is the density determining how many parameters are retained. We search for optimal values of λ within the range [0.5, 0.7, 1.0] and d within the range [0.2, 0.4, 0.6, 0.8].

DARE Yu et al. (2024a) also addresses the interference between parameters from different models during the model merging process. They randomly drop delta parameters with a probability of p and rescale the remaining ones by $1/(1-p)$. The DARE method can be combined with both the Task Arithmetic and TIES approaches. When combined with Task Arithmetic, a proportion p of task vectors is randomly dropped, and the remaining ones are rescaled by $1/(1-p)$. When DARE is combined with TIES, a proportion p of task vectors is randomly dropped, and the sign of each parameter is determined by comparing the total magnitude in the positive and negative directions. The sign corresponding to the larger total magnitude is selected, and only values matching this sign are retained. Their mean is then computed as the final parameter value, and the result is rescaled by $1/(1-p)$. The DARE method can also be expressed as:

$$\theta_{\text{trans}}^{\text{MERGE}} = \theta_{\text{trans}}^{\text{PRE}} + \lambda \cdot f(\tau^{\text{LVLM}}, d) + \lambda \cdot f(\tau^{\text{RM}}, d),$$

where d represents the density, determining the proportion of retained parameters, with $d = 1 - p$. We search for optimal values of λ within the range [0.5, 0.7, 1.0] and d within the range [0.2, 0.4, 0.6, 0.8].

Merging Embeddings We follow the embedding merging procedure from MergeKit (Goddard et al., 2024). The process is as follows:

1. If a token exists in the pre-trained model, we use its embedding from that model.
2. If a token appears in only one model (either the LVLM or the text-based RM), we use its embedding from that model.
3. If a token appears in multiple models, we compute the average of its embeddings.

Notably, the pre-trained model is not required for the weighted averaging method. Therefore, we omit the first step when applying this merging approach.

Merging Hyperparameter Selection We select the merging hyperparameter by using a sampled set of 400 instances from the RLAIF-V (Yu et al., 2024c) training set as our validation set. In case of a tie in scores, an additional 100 sampled instances will be used for evaluation. Results are discussed in Appendix G.3.

B Dataset Details

VL-RewardBench VL-RewardBench (Li et al., 2024a) is a benchmark comprising 1,250 high-quality examples spanning three domains: general multimodal instructions, hallucination-related tasks, and multimodal reasoning tasks. Each example includes a multimodal query—consisting of an image and a user prompt—along with a selected response and a rejected response.

TextVQA TextVQA (Singh et al., 2019) is a dataset designed to evaluate the ability of visual question-answering (VQA) models to read and reason about text within images. We use its validation set, which contains 5,000 instances, to assess our merged VLRMs.

MMMU-Pro MMMU-Pro (Yue et al., 2024b) is an advanced benchmark designed to assess the understanding and reasoning abilities of multimodal models. It is derived from the original MMMU (Yue et al., 2024a) dataset and consists of two subsets: a standard set, which includes image and text queries with 10 answer options, and a

vision set, which features a vision-only input scenario. In the vision set, the questions are embedded within screenshots or photos, with no explicit text provided.

RLAIF-V RLAIF-V (Yu et al., 2024c) preference dataset is created by generating multiple candidate responses for a given prompt and image using various random seeds. Each response is divided into individual claims, which are then assessed using an open-source large vision-language model. This model assigns confidence scores to each claim, which are combined to form an overall response score. Preference pairs are generated by comparing the response scores for the same prompt, selecting the preferred response and the less favorable one based on the score differences. Pairs with significant length disparities are excluded to avoid bias. We select 400 instances from this preference dataset to serve as our validation set for selecting the hyperparameters of merging methods.

C Best-of-N Sampling Details

We use lmms-eval (Zhang et al., 2024) for response generation with the Best-of-N sampling technique. For the TextVQA dataset, we set both the temperature and top-p to 1.0, sampling 8 responses. To encourage concise answers, we append "Answer the question using a single word or phrase." after the generation prompt. For the MMMU-Pro dataset, we also set the temperature and top p to 1.0, with a maximum token limit of 4096, to sample 8 responses. Additionally, we apply chain-of-thought (CoT) for generating both answers and their reasoning.

D Prompt Template

For Best-of-N sampling using `LLaMA-3.2-Vision` as the generative reward model, the prompt template is provided in Table 4. For image captioning with `LLaMA-3.2-Vision` and reward modeling using `Tulu-3-RM` and `Tulu-2.5-RM`, the detailed prompt template can also be found in Table 4.

E Open-Source Model Details

`Llama-3.2-11B-Vision-Instruct` `Llama-3.2-11B-Vision-Instruct` (Dubey et al., 2024) is an 11B-parameter LVLM consisting of three main components: a vision encoder, an adapter, and a pre-trained language model. The language model is based on `Llama-3.1-8B-Instruct`. The adapter incorporates cross-attention layers to integrate image representations into the language model. During adapter training, the language model remains frozen, enabling seamless drop-in replacement for Llama-3.1 series models without requiring retraining.

`Tulu-2.5-RM` `Tulu-2.5-RM` (Ivison et al., 2024) is a reward model initialized from `Llama-3.1-8B` and fine-tuned using the Tulu 2 recipe (Ivison et al., 2023). It is adapted for reward modeling by replacing the language modeling head with a linear layer and fine-tuning it on preference data from diverse sources, including Ultrafeedback (Cui et al., 2024), Nectar (Zhu et al., 2024), HH-RLHF (Bai et al., 2022), and AlpacaFarm (Dubois et al., 2023), among others.

`Tulu-3-RM` `Tulu-3-RM` (Lambert et al., 2024) is another reward model initialized from `Llama-3.1-8B` and fine-tuned following the Tulu 3 recipe (Lambert et al., 2024). Like `Tulu-2.5-RM`, it is adapted for reward modeling by replacing the language modeling head with a linear layer. However, `Tulu-3-RM` is trained on a mixture of on-policy and off-policy preference data collected through an enhanced version of the Ultrafeedback (Cui et al., 2024) pipeline. This dataset includes prompts from various sources, such as the SFT dataset in the Tulu 3 recipe, WildChat (Zhao et al., 2024), Ultrafeedback (Cui et al., 2024), and synthetic persona-augmented instructions.

F Qualitative Results

We investigate reward model behavior before and after merging, and we evaluate qualitatively on VL-RewardBench. Tables 5 and 6 present results for `Tulu-2.5-RM`, while Tables 7 and 8 show `Tulu-3-RM`. Red text indicates misalignment with the image. Before merging, the text-based reward model made incorrect predictions. After merging, the vision-language reward models correctly identified the better response. In most cases, more advanced merging methods—such as task arithmetic, TIES, and DARE—produce larger reward differences between chosen and rejected responses than simple weighted averaging.

G Full Results

G.1 Main Results

The main results of merging with Tulu-2.5-RM are discussed in Section 4.3 of the main text. As shown in Table 1, merged VLRMs consistently outperform Llama-3.2-Vision and Tulu-2.5-RM across nearly all merging methods and benchmarks. Notably, in VL-RewardBench, they show the greatest improvement in the Hallucination domain. In Best-of-N evaluation, they perform well in both TextVQA and MMMU-Pro. Additionally, merged VLRMs match or surpass the strong Cascade baseline, suggesting that merging captures more information than simply cascading two models.

A similar trend is observed when merging with Tulu-3-RM. As shown in Table 9, merged VLRMs outperform Llama-3.2-Vision and Tulu-3-RM across most methods and benchmarks. In VL-RewardBench, they improve mainly in the General and Hallucination domains. For Best-of-N evaluation, they perform well in MMMU-Pro, but only a few achieve results comparable to Llama-3.2-Vision in TextVQA, likely due to Tulu-3-RM's weaker performance in this task. While merging with Llama-3.2-Vision enhances performance over Tulu-3-RM, it does not surpass Llama-3.2-Vision's score. Additionally, merged VLRMs exceed the strong Cascade baseline in other benchmarks and remain competitive with it in TextVQA.

In Table 12, we compare our merged VLRMs with large open-source LVLMs and commercial systems on VL-RewardBench. Surprisingly, our merged VLRMs outperform 90B LVLMs and achieve performance comparable to commercial models, demonstrating the effectiveness of transferring textual preferences from text-based RMs to LVLMs.

G.2 Without Image Input

We conduct an ablation study by evaluating models without image input. Full results with Tulu-2.5-RM are shown in Table 10. Models with image input consistently outperform those without it across various merging techniques, suggesting that the vision encoder actively contributes after merging rather than performance gains being solely due to the text-based RM. This indicates that merged VLRMs effectively utilize the vision encoder in most cases. Notably, in VL-RewardBench, merged VLRMs match or surpass those without image input, especially in the hallucination domain, where image input significantly improves performance. In Best-of-N evaluation, models with image input perform better in the TextVQA and MMMU-Pro Vision sets. However, in the MMMU-Pro Standard set, image input does not provide an advantage, likely because this set emphasizes text reasoning, where reward assessments depend more on textual coherence than visual information.

Full results with Tulu-3-RM are shown in Table 11, following a similar trend. In VL-RewardBench, merged VLRMs outperform those without image input in the hallucination domain and are comparable to or surpass them in general and reasoning domains. Image input also enhances Best-of-N evaluation, particularly in TextVQA and MMMU-Pro Vision. However, in the MMMU-Pro Standard, image input does not provide a clear advantage, reaffirming that this set prioritizes text reasoning over visual input.

G.3 Effect of Merging Hyperparameters

In this study, we optimize hyperparameter merging using sampled instances from RLAIF-V. The results, based on 400 sampled RLAIF-V instances used as a validation set, are presented in Tables 13 to 22. Bold text highlights the best performance, while **text with *** indicates cases where scores are tied. In these cases, an additional 100 samples are used, and * marks the top-performing result among them.

Figures 3 to 12 show the effect of hyperparameters across various benchmarks, merging methods, and text-based RMs. The results reveal that optimal hyperparameters differ across these factors, emphasizing the importance of a well-constructed validation set. Future research could further explore this. For example, Figure 3 shows the results of searching for λ values between 0 and 1 for the Linear method using Tulu-2.5-RM. In the VL-RewardBench, a mid-range λ produces the best performance, while in the MMMU-Pro vision set, a smaller λ yields better results. This variation suggests that hyperparameter choices influence the performance of the final merged VLRMs differently depending on the task.

Moreover, we observe a trend consistent with prior studies (Yadav et al., 2024; Yu et al., 2024a): even when task vectors are reduced to lower rates (e.g., 0.4, 0.2), merged VLRMs continue to perform well, aligning with findings on LLM merging.

Best-of-N Sampling Prompt template for `Llama-3.2-Vision`
<|start_header_id|>system<|end_header_id|> You are a highly capable multimodal AI assistant tasked with evaluating answers to visual questions. <|eot_id|><|start_header_id|>user<|end_header_id|> Please analyze the following image and question, then evaluate the provided answer: Question: {INSTRUCTION} Answer: {RESPONSE} Evaluate the answer based on the following criteria: 1. Accuracy: How well does the answer align with the visual information in the image? Score: [1 (Poor) to 5 (Excellent)] 2. Completeness: Does the answer fully address all aspects of the question? Score: [1 (Poor) to 5 (Excellent)] 3. Clarity: Is the answer well-articulated and easy to understand? Score: [1 (Poor) to 5 (Excellent)] 4. Relevance: Does the answer directly relate to the question and the image? Score: [1 (Poor) to 5 (Excellent)] After your evaluation, please include: 1. Reasoning: A detailed explanation for each criterion, highlighting why you assigned the given score. 2. Overall Assessment: Provide a n overall quality score (1 to 5) for the answer. <|eot_id|>
Image Captioning Prompt template using `Llama-3.2-Vision`
<|start_header_id|>system<|end_header_id|> You are a highly capable multimodal AI assistant tasked with evaluating answers to visual questions. <|eot_id|><|start_header_id|>user<|end_header_id|> {IMAGE} Please describe this image according to the given question: {INSTRUCTION} <|eot_id|>
Prompt template for reward modeling
<|start_header_id|>user<|end_header_id|> {INSTRUCTION} <|eot_id|><|start_header_id|>assistant<|end_header_id|> {RESPONSE} <|eot_id|>

Table 4: Prompt template for best-of-n sampling, image captioning and reward modeling.

Image:

Query: Describe this image in detail.

Response 1:
The image showcases a lively scene of a couple of kids playing soccer on a field. They are kicking a sports ball around while several people, including young men and other onlookers, are watching and enjoying the game.

The spectators are spread throughout the scene, with some standing in different spots around the grassy area. There are also numerous chairs set up in the vicinity, possibly for the audience to take a rest or gather at a later time. Furthermore, various personal items, such as backpacks and a handbag, can be observed near the chairs.

Additionally, a few cars and an umbrella are present in the background, hinting at an outdoor, casual setting for this friendly soccer match.

Response 2:
The image showcases a lively scene of a couple of kids playing soccer on a field. They are kicking a sports ball around while several people, including young men and other onlookers, are watching and enjoying the game. The spectators are spread throughout the scene, with some standing in different spots around the grassy area. There are also numerous chairs set up in the vicinity, possibly for the audience to take a rest or gather at a later time. Moreover, the soccer goalposts stand tall, ready for the players to score. Additionally, the grassy terrain adds to the natural charm of the setting, making it an ideal place for a friendly soccer match.

Score comparison:

Method	Response 1 Score	Response 2 Score
Tulu-2.5-RM	2.17188	**2.27930**
Linear	**1.89551**	1.60156
Task Vec.	**3.50391**	1.82422
TIES	**2.67773**	1.96387
DARE + Task Vec.	**3.51758**	1.69141
DARE + TIES	**3.44336**	1.65430

Table 5: Qualitative results on VL-RewardBench using TULU-2.5-RM for merging.

Image:

Query: What is still on the tray?

Response 1:
There are still two portions of fries left on the tray.

Response 2:
There are still two hot dogs and some fries left on the tray.

Score comparison:

Method	Response 1 Score	Response 2 Score
Tulu-2.5-RM	0.56885	**1.20508**
Linear	**0.60986**	0.38477
Task Vec.	**2.47461**	0.55273
TIES	**1.76953**	1.10840
DARE + Task Vec.	**2.5918**	0.62988
DARE + TIES	**2.46484**	0.43701

Table 6: Qualitative results on VL-RewardBench using TULU-2.5-RM for merging.

Image:

Query: How many bananas are there in the image?

Response 1:
In the picture, there are at least 26 bananas. The bananas are divided into several bunches, which may indicate that they vary in size and ripeness.

Response 2:
There are 47 bananas in the picture. The bananas are divided into several bunches, which may indicate that their size and ripeness are different.

Score comparison:

Method	Response 1 Score	Response 2 Score
Tulu-3-RM	1.4043	**2.33594**
Linear	**1.33984**	0.51074
Task Vec.	**5.58203**	4.90234
TIES	**7.99219**	7.55469
DARE + Task Vec.	**4.35938**	2.88672
DARE + TIES	**5.51953**	5.20312

Table 7: Qualitative results on VL-RewardBench using TULU-3-RM for merging.

Image:

Query: Describe this image in detail.

Response 1:
In the image, there is a twin-propeller airplane parked on the tarmac, surrounded by a crowd of people. Some people seem to be disembarking from the plane, while others are walking around in the vicinity. There are at least 15 people in total, spread out near the airplane.

Luggage is being unloaded, with a suitcase visible near the center of the scene and a handbag located towards the right side. It appears that the passengers are either arriving at their destination or preparing to board the aircraft, as the luggage is being sorted and attended to.

Response 2:
In the image, the flight crew consisting of a pilot and co-pilot can be seen conversing near the twin-propeller airplane parked on the tarmac. A crowd of people surrounds them, with some individuals disembarking from the plane and others preparing to board. Ground support equipment such as baggage carts and fuel trucks are visible in the background near the airport terminal building. Luggage is being unloaded and sorted, with a suitcase and handbag visible in the foreground. The blue stripe on the airplane adds a pop of color to the scene. It appears that the passengers are either arriving at their destination or preparing to depart on their journey. An information sign can be seen towards the left side of the image.

Score comparison:

Method	Response 1 Score	Response 2 Score
Tulu-3-RM	3.94531	**4.74219**
Linear	**3.66016**	2.74414
Task Vec.	**5.23828**	2.99219
TIES	**7.72656**	5.67188
DARE + Task Vec.	**4.67188**	2.24414
DARE + TIES	**5.79688**	2.88477

Table 8: Qualitative results on VL-RewardBench using TULU-3-RM for merging.

Method	VL-RewardBench					TextVQA	MMMU-Pro	
	General	Hallucination	Reasoning	Overall	Macro Avg.	Overall	Standard	Vision
`Llama-3.2-Vision`	33.3*	38.4*	56.6*	42.9*	42.8*	46.4	28.8	19.8
`Tulu-3-RM`	45.4	36.6	56.6	43.0	46.2	27.4	29.4	20.4
`Random`	50.0	50.0	50.0	50.0	50.0	**48.2**	29.2	18.4
`Cascade`	54.1	40.5	57.2	46.7	50.6	38.3	31.3	**23.7**
`Linear`	47.5	51.0	55.0	51.5	51.2	45.8	29.1	19.0
`Task Vec.`	63.4	66.4	57.5	63.7	62.4	36.0	**31.6**	20.9
`TIES`	59.0	**74.1**	50.9	**66.0**	61.4	28.3	30.7	20.6
`DARE + Task Vec.`	63.4	68.9	**58.5**	65.4	**63.6**	36.1	30.2	20.9
`DARE + TIES`	**63.9**	65.6	57.2	63.2	62.2	56.9	31.4	21.8

Table 9: Comparison of merging methods across the VL-RewardBench, TextVQA, and MMMU-Pro datasets using `TULU-3-RM` for merging. *Indicates results from Li et al. (2024a).

Method	VL-RewardBench					TextVQA	MMMU-Pro	
	General	Hallucination	Reasoning	Overall	Macro Avg.	Overall	Standard	Vision
`Linear`	39.3 (-2.2)	52.3 (+20.8)	54.4 (-4.1)	51.0 (+11.2)	48.7 (+4.9)	54.7 (+8.9)	27.8 (-1.3)	22.1 (+0.5)
`w/o image input`	41.5	31.5	58.5	39.8	43.8	45.8	29.1	21.6
`Task Vec.`	48.6 (+4.3)	59.4 (+20.4)	59.7 (+0.6)	57.9 (+13.0)	55.9 (+8.4)	59.0 (+20.3)	31.0 (-0.8)	22.7 (+1.7)
`w/o image input`	44.3	39.0	59.1	44.9	47.5	38.7	31.8	21.0
`TIES`	43.7 (-1.1)	58.2 (+23.0)	58.5 (-0.6)	56.2 (+13.5)	53.5 (+7.1)	64.2 (+23.3)	29.1 (-2.1)	22.6 (+1.6)
`w/o image input`	44.8	35.2	59.1	42.7	46.4	40.9	31.2	21.0
`DARE + Task Vec.`	49.2 (+4.4)	61.7 (+23.4)	61.0 (+2.2)	59.7 (+15.2)	57.3 (+10.0)	58.8 (+22.6)	30.3 (-1.8)	22.4 (+1.6)
`w/o image input`	44.8	38.3	58.8	44.5	47.3	36.2	32.1	20.8
`DARE + TIES`	49.2 (+3.3)	59.1 (+19.2)	58.2 (-0.6)	57.4 (+11.8)	55.5 (+7.3)	57.3 (+20.4)	31.6 (-0.5)	22.0 (+1.2)
`w/o image input`	45.9	39.9	58.8	45.6	48.2	36.9	32.1	20.8

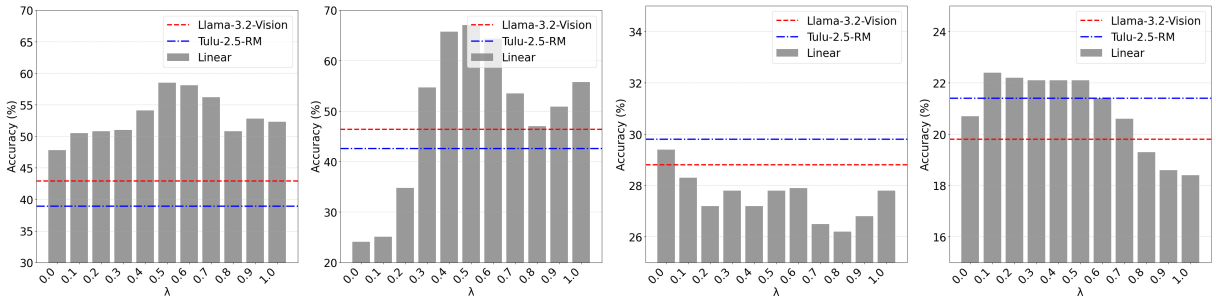
Table 10: Full results comparing merging methods with and without image input, using `TULU-2.5-RM` for merging.

Method	VL-RewardBench					TextVQA	MMMU-Pro	
	General	Hallucination	Reasoning	Overall	Macro Avg.	Overall	Standard	Vision
`Linear`	47.5 (-1.1)	51.0 (+1.1)	55.0 (0.0)	51.5 (+0.5)	51.2 (0.0)	45.8 (+25.5)	29.1 (+0.5)	19.0 (-1.3)
`w/o image input`	48.6	49.9	55.0	51.0	51.2	20.3	28.6	20.3
`Task Vec.`	63.4 (+3.8)	66.4 (+19.3)	57.5 (+4.4)	63.7 (+13.2)	62.4 (+9.1)	36.0 (+1.2)	31.6 (-0.1)	20.9 (+0.3)
`w/o image input`	59.6	47.1	53.1	50.5	53.3	34.8	31.7	20.6
`TIES`	59.0 (-0.6)	74.1 (+33.5)	50.9 (-3.2)	66.0 (+19.2)	61.4 (+10.0)	28.3 (-0.3)	30.7 (-1.0)	20.6 (-0.9)
`w/o image input`	59.6	40.6	54.1	46.8	51.4	28.6	31.7	21.5
`DARE + Task Vec.`	63.4 (+3.8)	68.9 (+18.4)	58.5 (+2.2)	65.4 (+12.1)	63.6 (+8.2)	36.1 (-5.8)	30.2 (-1.9)	20.9 (+0.7)
`w/o image input`	59.6	50.5	56.3	53.3	55.4	41.9	32.1	20.2
`DARE + TIES`	63.9 (+8.7)	65.6 (+20.9)	57.2 (+1.9)	63.2 (+14.2)	62.2 (+10.4)	56.9 (+29.2)	31.4 (+0.6)	21.8 (+1.4)
`w/o image input`	55.2	44.7	55.3	49.0	51.8	27.7	30.8	20.4

Table 11: Full results comparing merging methods with and without image input, using `TULU-3-RM` for merging.

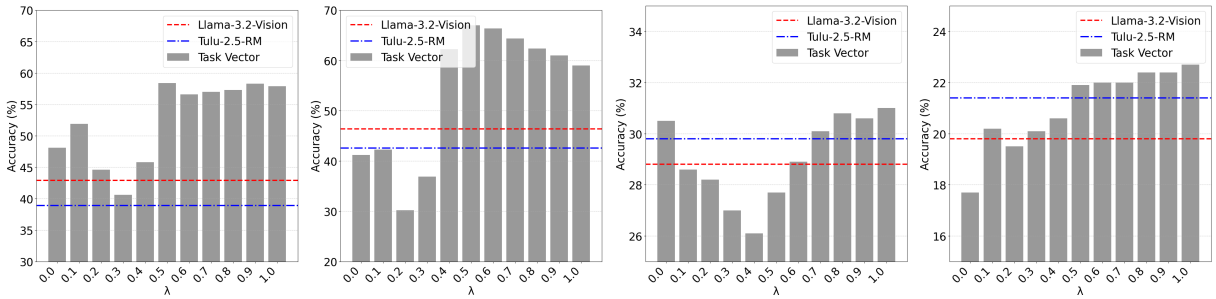
Method	General	Hallucination	Reasoning	Overall	Macro Avg.
*Open-Source Models**					
`Llama-3.2-Vision-11B-Instruct`	33.3	38.4	56.6	42.9	42.8
`Llama-3.2-Vision-90B-Instruct`	42.6	57.3	61.7	56.2	53.9
`Qwen2-VL-72B-Instruct`	38.1	32.8	58.0	39.5	43.0
`Molmo-72B-0924`	33.9	42.3	54.9	44.1	43.7
`NVLM-D-72B`	38.9	31.6	62.0	40.1	44.1
*Proprietary Models**					
`Gemini-1.5-Flash (2024-09-24)`	47.8	59.6	58.4	57.6	55.3
`Gemini-1.5-Pro (2024-09-24)`	50.8	72.5	64.2	**67.2**	62.5
`Claude-3.5-Sonnet (2024-06-22)`	43.4	55.0	62.3	55.3	53.6
`GPT-4o-mini (2024-07-18)`	41.7	34.5	58.2	41.5	44.8
`GPT-4o (2024-08-06)`	49.1	67.6	**70.5**	65.8	62.4
Using TULU-2.5-RM for merging					
`Linear`	39.3	52.3	54.4	51.0	48.7
`Task Vec.`	48.6	59.4	59.7	57.9	55.9
`TIES`	43.7	58.2	58.5	56.2	53.5
`DARE + Task Vec.`	49.2	61.7	61.0	59.7	57.3
`DARE + TIES`	49.2	59.1	58.2	57.4	55.5
Using TULU-3-RM for merging					
`Linear`	47.5	51.0	55.0	51.5	51.2
`Task Vec.`	63.4	66.4	57.5	63.7	62.4
`TIES`	59.0	**74.1**	50.9	66.0	61.4
`DARE + Task Vec.`	63.4	68.9	58.5	65.4	**63.6**
`DARE + TIES`	**63.9**	65.6	57.2	63.2	62.2

Table 12: Full results on VL-RewardBench, compared with current strong large vision-language models. *Indicates results from Li et al. (2024a).



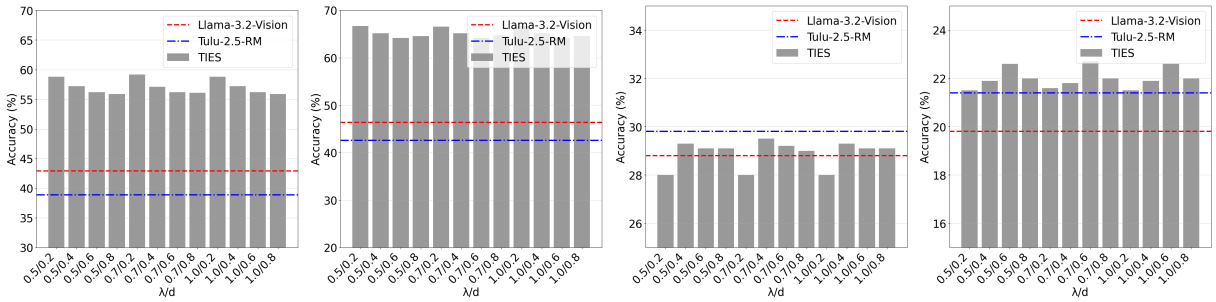
(a) VL-RewardBench (b) TextVQA (c) MMMU-Pro (Standard) (d) MMMU-Pro (Vision)

Figure 3: Full results of merging `Llama-3.2-Vision` and `Tulu-2.5-RM` (`Linear`)



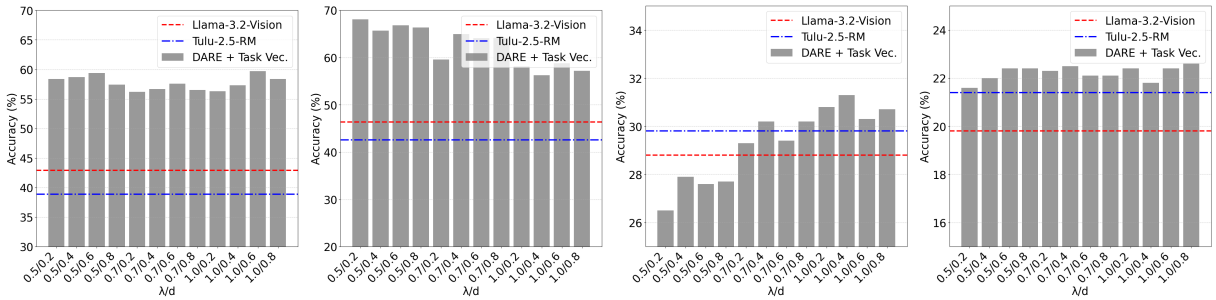
(a) VL-RewardBench (b) TextVQA (c) MMMU-Pro (Standard) (d) MMMU-Pro (Vision)

Figure 4: Full results of merging `Llama-3.2-Vision` and `Tulu-2.5-RM` (`Task Vec.`)



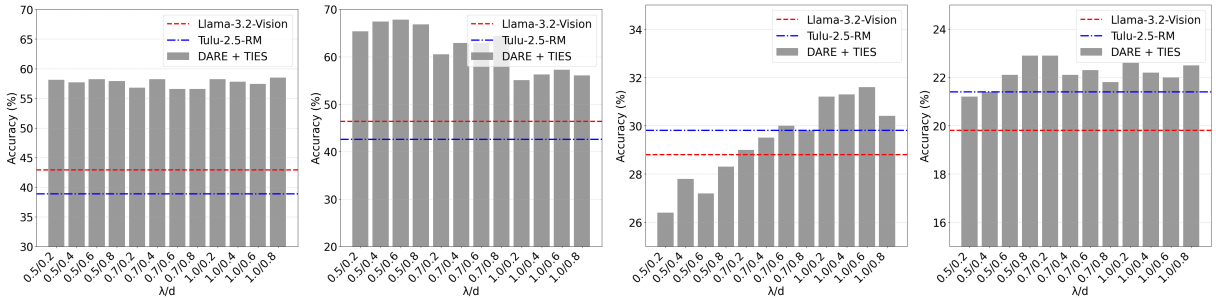
(a) VL-RewardBench (b) TextVQA (c) MMMU-Pro (Standard) (d) MMMU-Pro (Vision)

Figure 5: Full results of merging `Llama-3.2-Vision` and `Tulu-2.5-RM` (`TIES`)



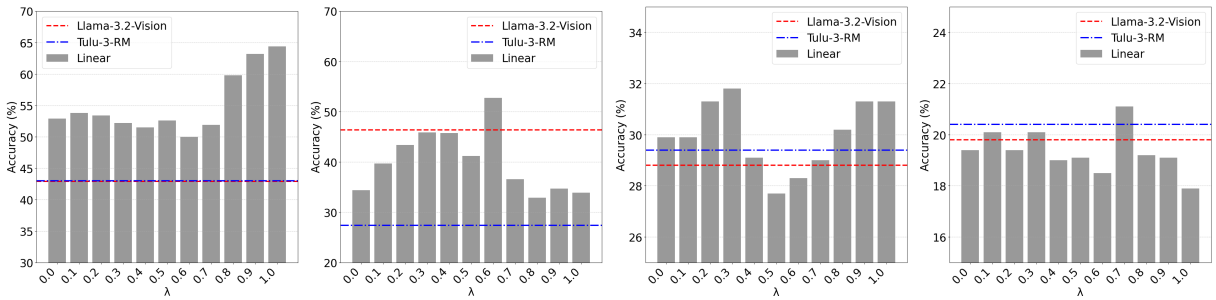
(a) VL-RewardBench (b) TextVQA (c) MMMU-Pro (Standard) (d) MMMU-Pro (Vision)

Figure 6: Full results of merging `Llama-3.2-Vision` and `Tulu-2.5-RM` (`DARE + Task Vec.`)



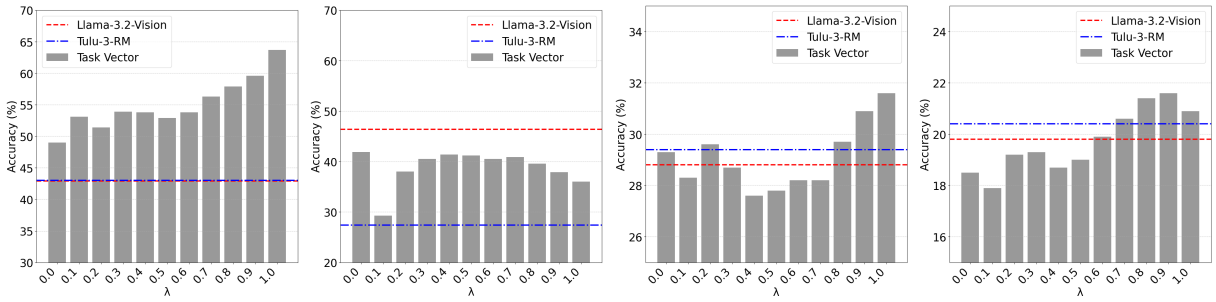
(a) VL-RewardBench (b) TextVQA (c) MMMU-Pro (Standard) (d) MMMU-Pro (Vision)

Figure 7: Full results of merging `Llama-3.2-Vision` and `Tulu-2.5-RM` (`DARE + TIES`)



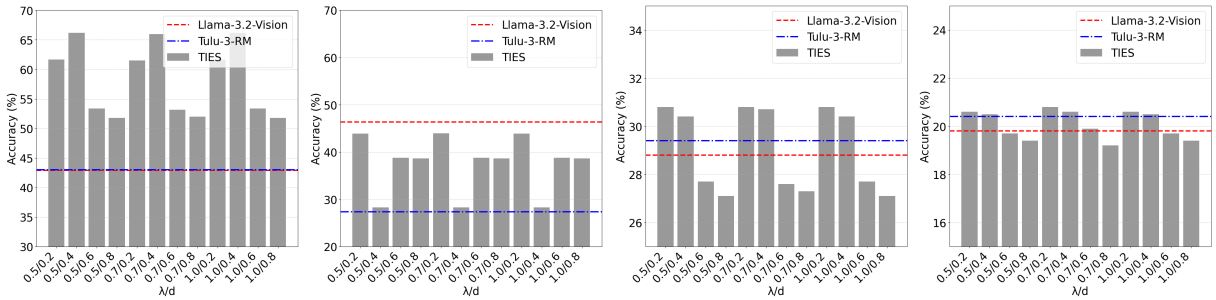
(a) VL-RewardBench (b) TextVQA (c) MMMU-Pro (Standard) (d) MMMU-Pro (Vision)

Figure 8: Full results of merging `Llama-3.2-Vision` and `Tulu-3-RM` (`Linear`)



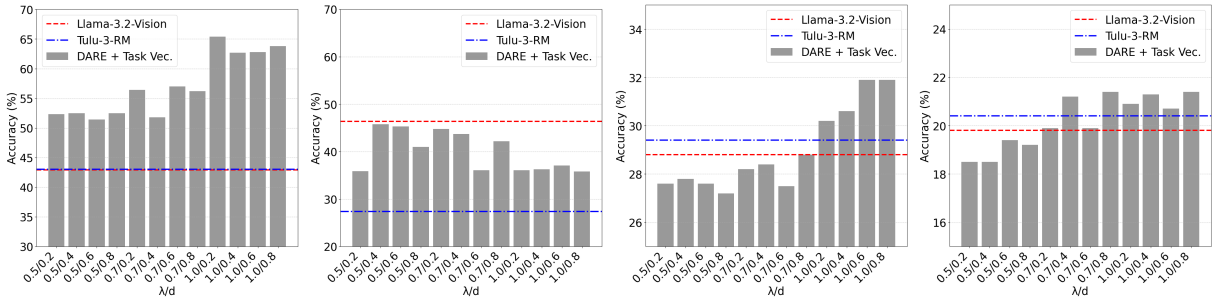
(a) VL-RewardBench (b) TextVQA (c) MMMU-Pro (Standard) (d) MMMU-Pro (Vision)

Figure 9: Full results of merging `Llama-3.2-Vision` and `Tulu-3-RM` (`Task Vec.`)



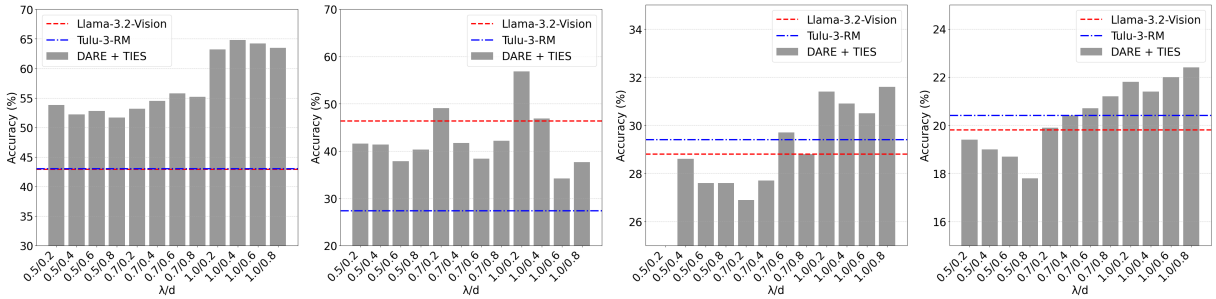
(a) VL-RewardBench (b) TextVQA (c) MMMU-Pro (Standard) (d) MMMU-Pro (Vision)

Figure 10: Full results of merging `Llama-3.2-Vision` and `Tulu-3-RM` (`TIES`)



(a) VL-RewardBench (b) TextVQA (c) MMMU-Pro (Standard) (d) MMMU-Pro (Vision)

Figure 11: Full results of merging `Llama-3.2-Vision` and `Tulu-3-RM` (`DARE + Task Vec.`)



(a) VL-RewardBench (b) TextVQA (c) MMMU-Pro (Standard) (d) MMMU-Pro (Vision)

Figure 12: Full results of merging `Llama-3.2-Vision` and `Tulu-3-RM` (`DARE + TIES`)

λ	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Overall Acc.	49.8	52.3	50.3	**52.5**	52.0	49.0	47.3	46.5	46.5	50.3	47.0

Table 13: `Linear` merging using `Tulu-2.5-RM` as the text-based RM, evaluated on sampled RLAIF-V.

λ	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Overall Acc.	**55.3**	50.0	53.3	54.5	53.5	49.3	52.8	54.0	53.8	54.8	**55.3***

Table 14: `Task Vec.` merging using `Tulu-2.5-RM` as the text-based RM, evaluated on sampled RLAIF-V.

λ	1.0				0.7				0.5			
d	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2
Overall Acc.	53.5	**53.8***	52.3	50.0	53.5	**53.8**	52.3	50.3	53.5	**53.8**	52.3	50.0

Table 15: `TIES` merging using `Tulu-2.5-RM` as the text-based RM, evaluated on sampled RLAIF-V.

λ	1.0				0.7				0.5			
d	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2
Overall Acc.	55.3	**56.5**	54.5	55.3	54.5	54.0	53.5	55.8	49.0	49.3	51.8	54.8

Table 16: `DARE + Task Vec.` merging using `Tulu-2.5-RM` as the text-based RM, evaluated on sampled RLAIF-V.

λ	1.0				0.7				0.5			
d	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2
Overall Acc.	55.5	**56.0***	**56.0**	55.5	53.3	54.3	53.8	52.3	51.5	49.8	51.5	51.8

Table 17: `DARE + TIES` merging using `Tulu-2.5-RM` as the text-based RM, evaluated on sampled RLAIF-V.

λ	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Overall Acc.	51.5	46.8	50.3	49.3	**52.0**	50.8	49.3	47.3	49.5	49.3	51.3

Table 18: `Linear` merging using `Tulu-3-RM` as the text-based RM, evaluated on sampled RLAIF-V.

λ	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Overall Acc.	49.3	53.5	49.8	49.8	51.0	51.0	53.8	53.0	53.0	50.3	**55.3**

Table 19: `Task Vec.` merging using `Tulu-3-RM` as the text-based RM, evaluated on sampled RLAIF-V.

λ	1.0				0.7				0.5			
d	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2
Overall Acc.	53.5	53.3	54.0	51.0	53.8	**54.3**	**54.3***	51.5	53.5	53.3	54.0	51.0

Table 20: `TIES` merging using `Tulu-3-RM` as the text-based RM, evaluated on sampled RLAIF-V.

λ	1.0				0.7				0.5			
d	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2
Overall Acc.	54.8	55.8	55.3	**58.0**	53.8	53.8	52.3	50.3	50.0	50.3	51.0	51.5

Table 21: `DARE + Task Vec.` merging using `Tulu-3-RM` as the text-based RM, evaluated on sampled RLAIF-V.

λ	1.0				0.7				0.5			
d	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2
Overall Acc.	55.8	55.8	56.0	**56.8**	52.8	52.5	52.5	52.3	55.3	53.8	48.0	54.5

Table 22: `DARE + TIES` merging using `Tulu-3-RM` as the text-based RM, evaluated on sampled RLAIF-V.

ProgCo: Program Helps Self-Correction of Large Language Models

Xiaoshuai Song[*], **Yanan Wu**[†], **Weixun Wang, Jiaheng Liu, Wenbo Su, Bo Zheng**
Taobao & Tmall Group of Alibaba
{songxiaoshuai.sxs,lixing.wyn}@alibaba-inc.com

Abstract

Self-Correction aims to enable large language models (LLMs) to self-verify and self-refine their initial responses without external feedback. However, LLMs often fail to effectively self-verify and generate correct feedback, further misleading refinement and leading to the failure of self-correction, especially in complex reasoning tasks. In this paper, we propose **Prog**ram-driven Self-**Co**rrection (**ProgCo**). First, **prog**ram-driven **ve**rification (**ProgVe**) achieves complex verification logic and extensive validation through self-generated, self-executing verification pseudo-programs. Then, **prog**ram-driven **re**finement (**ProgRe**) receives feedback from ProgVe, conducts dual reflection and refinement on both responses and verification programs to mitigate misleading of incorrect feedback in complex reasoning tasks. Experiments on three instruction-following and mathematical benchmarks indicate that ProgCo achieves effective self-correction, and can be further enhance performance when combined with real program tools. We release our code at https://github.com/songxiaoshuai/progco.

1 Introduction

Although large language models (LLMs) have shown excellent performance on certain tasks, they still face challenges such as hallucinations and unfaithful reasoning when solving complex instruction-following and reasoning tasks (Zhao et al., 2023; Chang et al., 2024). Self-correction is an expected capability of LLMs, wherein the LLM first needs to reflect on its initial output, identify potential issues and generate feedback (*self-verification* phase), which then guides the LLM to optimize and refine its output (*self-refinement* phase), as illustrated in Fig 1 (Pan et al., 2024; Kumar et al., 2024). However, studies have shown

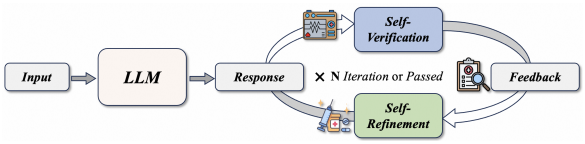


Figure 1: Illustration of a typical workflow of LLM's intrinsic self-correction.

that current LLMs severely lack this capability and struggle to achieve effective self-correction in the absence of external feedback (also called *intrinsic self-correction*), particularly in complex tasks (Li et al., 2024; Huang et al., 2024; Tyen et al., 2024; Kamoi et al., 2024).

A core reason for the failure of self-correction in LLMs is their inability to effectively self-detect problematic outputs and generate high-quality feedback (Tyen et al., 2024). Existing works mainly used two approaches for self-verification: (1) prompting LLMs to perform step-by-step self-checks (Madaan et al., 2024), or (2) generating a checklist based on the task and then checking responses against this list (Zhang et al., 2024c; Cook et al., 2024). However, on one hand, LLMs often exhibit overconfidence, making it difficult for them to identify their own errors or hallucinations, resulting in a low recall of incorrect responses (Zhang et al., 2024c). On the other hand, for complex tasks, these methods struggle to parse intricate verification logic. For instance, checklists usually only express parallel relationships and examine superficial issues. Ineffective verification further leads to ineffective refinement. The inaccurate and low-quality error detection and feedback not only makes it challenging for LLMs to correct erroneous outputs in one attempt but also seriously mislead them into modifying from correct to incorrect for misrecalled responses. To overcome this issue, we propose **Prog**ram-driven Self-**Co**rrection (**ProgCo**), achieving effective self-correction by incorporating self-generated and self-executed programs in the

[*]Work done during internship at Alibaba Inc.
[†]Corresponding Author: Yanan Wu

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 944–959
July 27 - August 1, 2025

verification and refinement stages.

We first introduce **Prog**ram-driven **Ve**rification (**ProgVe**) to achieve better self-verification. Different from studies like PAL and POT (Gao et al., 2023; Chen et al., 2023; Gou et al., 2024), which integrate program or symbolic solvers for forward reasoning, ProgVe focuses on the self and reverse verification phase. In general, the LLM is initially prompted to generate a pseudo verification program for the input task. After obtaining the initial response, the LLM further acts as a program-executor, executing the verification program step by step to obtain the verification results. The motivation for ProgVe stems from: (1) Compared to the ambiguity of natural language, code program can express more complex verification logic and structures. (2) Using LLM as a program-executor not only allows a focus on the verification logic of the code without requiring strict executability, but also incorporate LLM's own knowledge and causal understanding into execution, such as the virtual function `is_structured_as_letter` in Fig 2.

To address misleading self-refinement caused by incorrect self-verification in complex reasoning tasks such as mathematics, we further introduce **Prog**ram-driven **Re**finement (**ProgRe**), featuring a dual refinement mechanism for both response and program. Within the framework, to avoid directly misleading of feedback, the response revised based on feedback is treated as an intermediate and contrasted with the pre-revision response to identify differences and generate insights, which will help in regenerating the final refined response. To more fundamentally address incorrect verification, besides refining response with grogram, the verification program is also reflected upon and optimized with the assistance of response information.

We provide a detailed exposition of proposed method in Section 2. In Section 3, we demonstrate that our method can achieve effective self-correction and outperforms all baselines in correcting instruction-following and mathematical reasoning tasks. A series of analyses further provides insights into self-correction and our method from multiple perspectives.

In summary, we propose ProgCo for effective self-correction, consisting of two components: ProgVe and ProgRe. Our contributions are threefold: (1) We propose ProgVe, a method enabling LLMs to self-generate and self-execute validation programs for self-verification. (2)We propose ProgRe, a self-refinement method that is robust to incorrect feedback and combines dual optimization of response and verification programs. (3) Experiments and analyses on three datasets verify the effectiveness of our method in self-correction.

Instruction: Write a letter to a friend in all lowercase letters ask them to go and vote.

```
def validate_response(response):
    errors = []
    if not response.islower():
        errors.append("The response is not in all lowercase letters.")
    if not is_structured_as_letter(response):
        errors.append("The response is not structured as a letter.")
    ……
    if errors:
        return False, "\n".join(errors)
    return True, "No Error"
```

Question: Kyle bought last year's best-selling book for $19.50. This is with a 25% discount from the original price. What was the original price of the book?

```
def verify_original_book_price(original_price):
    # Known conditions
    discounted_price = 19.50
    discount_percentage = 0.25

    # Calculate the discount amount
    discount_amount = original_price * discount_percentage

    # Calculate the price after discount
    calculated_discounted_price = original_price - discount_amount

    # Check if the calculated discounted price matches the known price
    if calculated_discounted_price != discounted_price
        return False

    return True
```

Figure 2: Illustration of generating verification pseudo-program for input tasks.

2 Method

2.1 Program-driven Verification

Given model M and input x, in the i-th iteration of self-correction, the goal of self-verification is to generate feedback fb_i for response y_i, indicating whether y_i passes verification or the reason it fails.

Verification Program Generation. After obtaining the initial response $y_0 = M(x)$, we first use prompt P_{fuc}^{gen} to guide M in generating a verification pseudo-program function f. This process is independent of y_0 to ensure a different perspective and avoid biases from the response:

$$f = M(P_{fuc}^{gen}||x) \tag{1}$$

As illustrated in Fig 2, for instruction-following task, f verifies a series of constraints extracted from x. For mathematical problems, f starts from the output answer and uses reverse reasoning to verify step-by-step whether it contradicts the given conditions in x.

Verification Program Execution. For each round, we use prompt P_{fuc}^{exec} to instruct M act as a code executor, taking y_i as input, executing f step by step to obtain the execution result r_i. r_i is further converted to feedback fb_i by prompt P_{fb}:

$$r_i = M(P_{fuc}^{exec}||x||y_i), fb_i = M(P_{fb}||x||y_i||r_i) \tag{2}$$

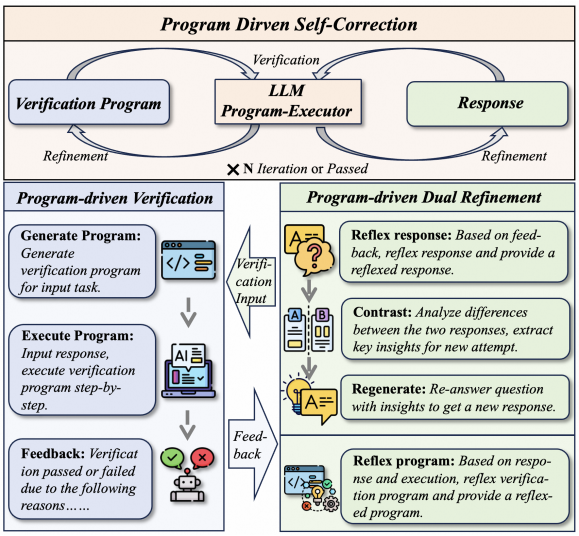


Figure 3: The overall framework of ProgCo, achieving self-correction through iterative ProgVe and ProgRe.

If the verification passes, self-correction stops and yields the final output $y_{final} = y_i$. Otherwise, if fb_i indicates that y_i fails to meet constraints or contains contradictions, the process enters the self-refinement stage until maximum rounds I are reached.

2.2 Program-driven Refinement

The **vanilla self-refinement** generates a new output $y_{i+1} = M(p_{refine}||x||y_i||fb_i)$ through prompt p_{refine}. This method is effective for tasks like instruction following, as constraints are apparent, and the correct program f and clear feedback fb_i are easily obtained. However, in reasoning tasks such as mathematics, the correct f and fb_i are difficult to obtain in one attempt and will mislead refinement, as described in Section 1. Therefore, we introduce the Program-driven Refinement framework to address this, as shown in Fig 3.

Preliminary Reflection. Unlike directly refining y_i using fb_i, we prompt M to first reflect on y_i under the feedback and then output a temporary response y_{i+1}^{temp}, which can either maintain y_i unchanged or update it:

$$y_{i+1}^{temp} = M(P_{reflex}||x||y_i||fb_i) \tag{3}$$

Contrast and Regenerate. If a new answer is obtained, y_{i+1}^{temp} is further compared with y_i to identify differences. These differences will be transformed into insights ins for problem-solving, assisting in regenerating the refined response:

$$ins = M(P_{cont}||y_i||y_{i+1}^{temp}), y_{i+1} = M(ins||x) \tag{4}$$

Verification Program Refinement. To correct potentially incorrect validation code generated in the initial round, we further introduce self-refinement of the validation code. Utilizing information from y_i and fb_i, the validation code f_i will self-reflect and generate new validation code f_{i+1} for next round of self-verification:

$$f_{i+1} = M(P_{reflex}^{code}||x||f_i||y_i||fb_i) \tag{5}$$

We provide detailed pseudo-code and prompts used in the proposed method in Appendix B.

3 Experiment

3.1 Experiment Setup

We evaluate ProgCo on the instruction-following dataset IFEval (Zhou et al., 2023) and the mathematics datasets GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021)[1]. Since complex reasoning is not involved, we only use the combination of ProgVe and vanilla refinement for IFEval. Additionally, we calculate the score for each self-correction round individually, without assuming the next round is correct just because the previous one was. This may differ from some evaluation settings. We provide detailed datasets, baseline introductions, and implementation details in Appendix C.

3.2 Main Result

Table 1 shows the performance comparison between ProgCo and different self-correction baselines. [2] Overall, ProgCo outperforms all baselines with a large margin across three benchmarks. On GPT-3.5, ProgCo improves over the initial response by 4.62% (IFEval(Pr)), 3.23% (IFEval(Ins)), 5.84% (GSM8K), and 5.8% (MATH) with just one round of self-correction. After three rounds, the improvements further increase to 4.80% (IFEval(Pr)), 3.47% (IFEval(Pr)),7.28% (GSM8K) and 8.0% (MATH). Similar improvements are observed on other LLMs. From a task perspective, many baselines achieve positive improvements on IFEval but failed on mathematical tasks, while our method achieve significant positive improvements on both

[1]We randomly sample 500 instances from MATH test set.

[2]Due to varied focuses, we discuss various inference methods in Appendix A and compare the self-consistency sampling methods in Section 3.6.

Method	Llama3.1-8B-Instruct				GPT-3.5				GPT-4o			
	IF (Pr)	IF (Ins)	GSM8K	MATH	IF (Pr)	IF (Ins)	GSM8K	MATH	IF (Pr)	IF (Ins)	GSM8K	MATH
Initial Score	73.75	81.29	85.82	46.8	58.23	68.35	76.5	36.2	82.99	87.52	95.07	77.2
Maximum One Round of Self-Correction												
Vanilla-reflex	68.39$_{-5.36}$	76.74$_{-4.55}$	83.32$_{-2.50}$	44.6$_{-2.20}$	54.9$_{-3.33}$	69.18$_{+0.83}$	72.4$_{-4.10}$	35.6$_{-0.60}$	83.73$_{+0.74}$	88.37$_{+0.85}$	93.31$_{-1.76}$	77.6$_{+0.40}$
Self-Refine	75.79$_{+2.04}$	82.49$_{+1.20}$	84.69$_{-1.13}$	40.6$_{-6.20}$	59.33$_{+1.10}$	68.94$_{+0.59}$	76.35$_{-0.15}$	38.0$_{+1.80}$	84.66$_{+1.67}$	88.97$_{+1.45}$	94.92$_{-0.15}$	75.4$_{-1.80}$
Self-Reflection	73.38$_{-0.37}$	80.22$_{-1.07}$	84.23$_{-1.59}$	43.8$_{-3.00}$	59.7$_{+1.47}$	68.94$_{+0.59}$	71.95$_{-4.55}$	34.8$_{-1.40}$	84.84$_{+1.85}$	88.97$_{+1.45}$	95.0$_{-0.07}$	77.6$_{+0.40}$
CheckList	74.49$_{+0.74}$	81.89$_{+0.60}$	84.76$_{-1.06}$	47.0$_{+0.20}$	59.15$_{+0.92}$	69.06$_{+0.71}$	77.63$_{+1.13}$	36.0$_{-0.20}$	85.4$_{+2.41}$	89.45$_{+1.93}$	94.84$_{-0.23}$	77.4$_{+0.20}$
ProgCo (Ours)	**76.34**$_{+2.59}$	**83.69**$_{+2.40}$	**86.58**$_{+0.76}$	**50.2**$_{+3.40}$	**62.85**$_{+4.62}$	**71.58**$_{+3.23}$	**82.34**$_{+5.84}$	**42.0**$_{+5.80}$	**87.99**$_{+5.00}$	**91.85**$_{+4.33}$	**95.75**$_{+0.68}$	**79.4**$_{+2.20}$
Maximum Three Rounds of Self-Correction												
Vanilla-reflex	65.25$_{-8.50}$	74.82$_{-6.47}$	81.43$_{-4.39}$	44.6$_{-2.20}$	55.27$_{-2.96}$	65.83$_{-2.52}$	71.8$_{-4.70}$	35.8$_{-0.40}$	84.29$_{+1.30}$	88.73$_{+1.21}$	94.08$_{-0.99}$	79.2$_{+2.00}$
Self-Refine	76.34$_{+2.59}$	82.73$_{+1.44}$	84.99$_{-0.83}$	43.4$_{-3.40}$	59.7$_{+1.47}$	69.06$_{+0.71}$	76.35$_{-0.15}$	38.0$_{+1.80}$	85.03$_{+2.04}$	89.45$_{+1.93}$	94.92$_{-0.15}$	76.4$_{-0.80}$
Self-Reflection	73.75$_{+0.00}$	81.06$_{-0.23}$	84.53$_{-1.29}$	48.0$_{+1.20}$	59.52$_{+1.29}$	69.18$_{+0.83}$	74.07$_{-2.43}$	35.6$_{-0.60}$	85.77$_{+2.78}$	89.69$_{+2.17}$	94.92$_{-0.15}$	78.4$_{+1.20}$
CheckList	75.05$_{+1.30}$	82.61$_{+1.32}$	84.91$_{-0.91}$	47.4$_{+0.60}$	59.52$_{+1.29}$	69.3$_{+0.95}$	77.79$_{+1.29}$	35.8$_{-0.40}$	85.77$_{+2.78}$	89.69$_{+2.17}$	95.0$_{-0.07}$	77.0$_{-0.20}$
ProgCo (Ours)	**77.82**$_{+4.07}$	**84.29**$_{+3.00}$	**87.41**$_{+1.59}$	**50.6**$_{+3.80}$	**63.03**$_{+4.80}$	**71.82**$_{+3.47}$	**83.78**$_{+7.28}$	**44.2**$_{+8.00}$	**87.8**$_{+4.81}$	**91.97**$_{+4.45}$	**95.75**$_{+0.68}$	**80.0**$_{+2.80}$

Table 1: The result of different self-correction methods. The metric for GSM8K and MATH is accuracy, IF (Pr) and IF (Ins) denote IFEval's strict prompt and instruction metrics, respectively. +/- indicates change from initial score.

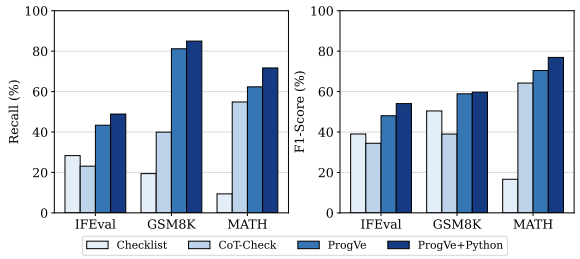


Figure 4: Recall and F1 scores of self-verification methods for incorrect responses on GPT-3.5.

GSM8K and MATH, demonstrating its effectiveness in complex reasoning.

3.3 Ablation Analysis

Recall of Self-Verification. Fig 4 shows the recall and F1-score for CoT-Check, Checklist, and ProgVe in identifying incorrect responses. ProgVe outperforms the baselines in both Recall and F1-score significantly, with further improvements when combined with Python tool. This demonstrates the advantages of using self-play programs in verification, including expressing more complex structures, providing a different perspective, and integrate with symbolic tools.

Ablation on Self-Refinement. We conduct an ablation analysis of ProgRe in Table 2. First, removing contrast and regeneration leads to a significant increase in the ratio of correction to incorrection, proving that this strategy can effectively alleviate feedback misleading. Second, since wrong verification programs continually trap responses from passing verification, removing the reflection of program leads to a significant increase in the average rounds of self-correction. Lastly, without program feedback involved in response reflection,

Method	ACC ↑	Avg Turn ↓	$\Delta^{i\to c}$ ↑	$\Delta^{c\to i}$ ↓
ProgVe	83.78	0.88	47.75	15.42
w/o cont. & regen.	79.08$_{-4.7}$	0.87$_{-0.01}$	43.88$_{-3.87}$	35.18$_{+19.76}$
w/o program reflex	83.02$_{-0.76}$	1.06$_{+0.18}$	44.98$_{-2.77}$	15.95$_{+0.53}$
w/o feedback	81.35$_{-2.43}$	0.90$_{+0.02}$	30.5$_{-17.25}$	7.36$_{-8.06}$

Table 2: Ablation on GPT-3.5 with maximum three rounds of self-correction. Avg Turn is the average rounds of self-refinement for all samples. $\Delta^{i\to c}$ is the ratio of incorrect-to-correct transitions among recalled ground-truth incorrect samples, and $\Delta^{c\to i}$ is the opposite for recalled correct samples. Avg Turn denotes the average ProgVe turns for all samples.

Model	IFEval (Pr)		MATH (ACC)	
	ProgVe	ProgVe+Python	ProgVe	ProgVe+Python
GPT-3.5	62.0	**64.14**$_{+2.14}$	44.2	**44.6**$_{+0.4}$
GPT-4o	87.8	**91.31**$_{+3.51}$	80.0	**81.2**$_{+1.2}$

Table 3: Performance of ProgCo in integrating the Python executor tool during the ProgVe process.

the ratios of both correct and incorrect refinement notably decrease, indicating issues of LLM's overconfidence and misleading feedback, respectively.

3.4 Integrating symbolic tools for LLM program-executor.

Due to the advantage of program easily integrating with symbolic tools, we further indicate in Prompt P_{fuc}^{exec} that LLM executor can delegate complex numerical operations to an actual python tool to overcome the shortcomings of LLM. As shown in Table 3, this further improves ProgCo's performance. For example, ProgVe obtains more precise feedback on constraints such as word count and keywords, thereby significantly improving the performance of IFEval.

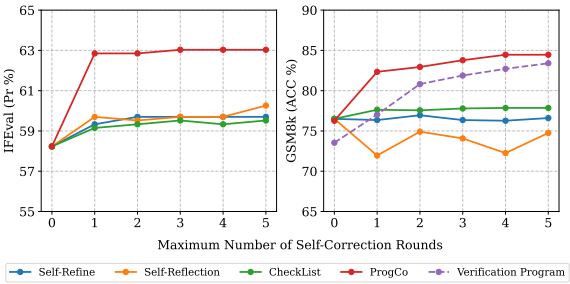


Figure 5: Score variation with the maximum number of self-correction rounds on GPT-3.5. The program's ACC measures the consistency accuracy between the execution results of ProgVe and ground-truth scores.

Method	MAX Turn	IFEval(Pr)	GSM8K	MATH
Initial Score	–	58.23	76.5	36.2
SC-Vote	3	–	79.45	37.4
	5	–	83.02	40.8
	10	–	**84.46**	44.2
SC-Reflex	3	41.96	79.61	38.2
	5	42.88	81.73	40.4
	10	44.36	84.08	41.6
SC-Select	3	55.82	77.41	36.2
	5	55.64	78.24	38.6
	10	56.19	78.09	36.0
Ours	1	62.85 (0.34 turn_{avg})	82.34 (0.41 turn_{avg})	42.0 (0.49 turn_{avg})
	3	63.03 (0.80 turn_{avg})	83.78 (0.88 turn_{avg})	44.2 (0.97 turn_{avg})
	5	**63.03** (1.18 turn_{avg})	**84.46** (1.13 turn_{avg})	**44.4** (1.34 turn_{avg})

Table 4: Comparison of ProgCo and the Self-Consistency (SC) on GPT-3.5. The temperature for SC is 0.7. Turn_{avg} denotes the average self-refinement (ProgRe) iterations for all samples.

3.5 Effect of Self-Correction Iterations.

As shown in Fig 5, with increasing self-correction rounds, baselines' performance either slightly improves or fluctuates near the initial score. In contrast, ProgCo shows a consistent upward trend, especially in the first few rounds. The continual improvement in the verification program's consistency accuracy further highlights the advantages of dual refinement.

3.6 Comparison with Self-Consistency

We choose the self-consistency (SC) as the sampling approach. For multiple sampled responses, SC-Vote determines the final answer based on majority voting, SC-Reflex reflects to arrive at a final response, and SC-Select prompts the LLM to compare and select one as the final response. The results are shown in Table 4. On IFEval, SC-Vote is not applicable, and both SC-Reflex and SC-Select lead to poorer performance, indicating that sampling is not suitable for tasks like instruction-following. In contrast, ProgCo significantly improves instruction-following performance. On mathematical tasks where sampling excels, ProgCo with one correction round outperforms 3-sampling significantly and 5-sampling on MATH. ProgCo with five correction rounds matches 10-sampling on GSM8K and slightly surpasses it on MATH. It should be emphasized that, due to early stopping from ProgVe, ProgRe only performs an average of about 1.2 rounds of self-refinement (for instance, many cases pass self-verification without entering the self-refinement phase).

3.7 Case Study

We present examples of ProgVe performing self-verification on IFEval and GSM8K in Figures 12 and 13, respectively. Figure 14 shows an example of PogRe performing self-refinement on GSM8K. In Fig 12, the verification function not only generates executable programs such as `response!=response.upper()`, but also generates more vaguely defined programs like `is_english(response)` and `has_title(response)`. However, the LLM is able to successfully execute these pseudo-programs, surpassing the capabilities of a real program executor. In Fig 13, the verification program verifies the correctness of the response in reverse. It starts with the predicted `remaining_speed`, calculates the `average_speed` step by step, and checks whether it matches the given condition `target_average_speed`. In Fig 14, although the response still contains errors upon initial reflection, the LLM ultimately provides the correct answer to the question with the help of contrasted insights. Meanwhile, the verification program is also further optimized during the refinement process.

4 Conclusion

In this paper, we propose ProgCo, a program-driven self-correction method. ProgCo first self-generates and self-executes verification pseudo-programs for self-verification (ProgVe), then uses dual reflection and refinement of responses and programs for self-refinement (ProgRe). Experiments and analyses on three benchmarks demonstrate the effectiveness of ProgCo in self-correction.

Limitations

In this paper, we propose ProgCo and experimentally validate its effective self-correction. However, there are still some limitations as follows: (1) In terms of application scenarios, although using pseudo-program and LLM executors can extend the application scope beyond numerical and symbolic solving tasks, we primarily validated the effectiveness of ProgCo in instruction-following and mathematical tasks. (2) One advantage of using LLMs in executing verification programs is the integration of their own knowledge and causal logic. However, they are limited in large and precise numerical calculations. This issue can be mitigated by combining real symbolic tools, as shown in the experiment in Table 3. (3) Due to the lack of specialized training, we use detailed prompts to guide the LLM in completing tasks in ProgCo, which results in additional inference costs. Synthesizing data for each component of ProgCo and jointly training the LLM can replace the need for prompts and demonstration costs during inference.

References

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2024. Teaching large language models to self-debug. In *The Twelfth International Conference on Learning Representations*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Jonathan Cook, Tim Rocktäschel, Jakob Foerster, Dennis Aumiller, and Alex Wang. 2024. Ticking all the boxes: Generated checklists improve llm evaluation and generation. *Preprint*, arXiv:2410.03608.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578, Bangkok, Thailand. Association for Computational Linguistics.

Guanting Dong, Keming Lu, Chengpeng Li, Tingyu Xia, Bowen Yu, Chang Zhou, and Jingren Zhou. 2024a. Self-play with execution feedback: Improving instruction-following capabilities of large language models. *Preprint*, arXiv:2406.13542.

Guanting Dong, Xiaoshuai Song, Yutao Zhu, Runqi Qiao, Zhicheng Dou, and Ji-Rong Wen. 2024b. Toward general instruction-following alignment for retrieval-augmented generation. *arXiv preprint arXiv:2410.09584*.

Guanting Dong, Chenghao Zhang, Mengjie Deng, Yutao Zhu, Zhicheng Dou, and Ji-Rong Wen. 2024c. Progressive multimodal reasoning via active retrieval. *arXiv preprint arXiv:2412.14835*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: Program-aided language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10764–10799. PMLR.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2024. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In *The Twelfth International Conference on Learning Representations*.

Haixia Han, Jiaqing Liang, Jie Shi, Qianyu He, and Yanghua Xiao. 2024. Small language model can self-correct. *Preprint*, arXiv:2401.07301.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2024. Language models can solve computer tasks. *Advances in Neural Information Processing Systems*, 36.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2024. Training language models to self-correct via reinforcement learning. *Preprint*, arXiv:2409.12917.

Yanhong Li, Chenghao Yang, and Allyson Ettinger. 2024. When hindsight is not 20/20: Testing limits on reflective thinking in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3741–3753, Mexico City, Mexico. Association for Computational Linguistics.

Chenyang Lyu, Lecheng Yan, Rui Xing, Wenxi Li, Younes Samih, Tianbo Ji, and Longyue Wang. 2024. Large language models as code executors: An exploratory study. *Preprint*, arXiv:2410.06667.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Meta. 2024. Introducing llama 3.1: Our most capable models to date.

OpenAI. 2024. Hello gpt-4o.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2024. Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies. *Transactions of the Association for Computational Linguistics*, 12:484–506.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, et al. 2024a. We-math: Does your large multimodal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*.

Shuofei Qiao, Honghao Gui, Chengfei Lv, Qianghuai Jia, Huajun Chen, and Ningyu Zhang. 2024b. Making language models better tool learners with execution feedback. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3550–3568, Mexico City, Mexico. Association for Computational Linguistics.

Matthew Renze and Erhan Guven. 2024. Self-reflection in llm agents: Effects on problem-solving performance. *arXiv preprint arXiv:2405.06682*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. 2024. LLMs cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13894–13908, Bangkok, Thailand. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhenyu Wu, Qingkai Zeng, Zhihan Zhang, Zhaoxuan Tan, Chao Shen, and Meng Jiang. 2024. Large language models can self-correct with key condition verification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12846–12867, Miami, Florida, USA. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024a. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Wenlin Yao, Haitao Mi, and Dong Yu. 2024b. Hdflow: Enhancing llm complex problem-solving with hybrid thinking and dynamic workflows. *arXiv preprint arXiv:2409.17433*.

Che Zhang, Zhenyang Xiao, Chengcheng Han, Yixin Lian, and Yuejian Fang. 2024a. Learning to check: Unleashing potentials for self-correction in large language models. *Preprint*, arXiv:2402.13035.

Di Zhang, Xiaoshui Huang, Dongzhan Zhou, Yuqiang Li, and Wanli Ouyang. 2024b. Accessing gpt-4 level mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b. *arXiv preprint arXiv:2406.07394*.

Wenqi Zhang, Yongliang Shen, Linjuan Wu, Qiuying Peng, Jun Wang, Yueting Zhuang, and Weiming Lu. 2024c. Self-contrast: Better reflection through inconsistent solving perspectives. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3602–3622, Bangkok, Thailand. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

A Related Work

Self-Correction. Self-correction aims to enable LLMs to achieve the ability to self-check and correct its outputs (Pan et al., 2024). Although some work (Shinn et al., 2024; Renze and Guven, 2024) achieves correction by relying on environmental feedback (such as True/False signals), many studies (Li et al., 2024; Huang et al., 2024; Tyen et al., 2024; Kamoi et al., 2024) have shown that in the complete absence of environmental feedback, LLMs find it difficult to engage in effective self-reflection, with a particular emphasis on the challenge of identifying their own errors. To this end, some work (Han et al., 2024; Zhang et al., 2024a; Kumar et al., 2024) focuses on enhancing LLMs' self-correction capabilities during the training phase through imitation learning or reinforcement learning, while another part (Madaan et al., 2024; Dhuliawala et al., 2024; Wu et al., 2024; Zhang et al., 2024c; Kim et al., 2024) focuses on designing reflection or correction pipelines during the inference phase to help LLMs examine and analyze their own outputs. In this work, we focus on the inference phase, and to the best of our knowledge, we are the first to introduce self-generated and self-executed pseudo-verification programs into self-verification and self-refinement, achieving effective self-correction.

Integration of LLM with Programs. Several studies have enhanced LLMs by introducing programs or symbolic solvers. Some works integrate code executor or symbolic solving tools within the LLM's forward reasoning to address mathematical or symbolic reasoning problems, such as PAL (Gao et al., 2023), POT (Chen et al., 2023), and ToRA (Gou et al., 2024). Others (Chen et al., 2024; Dong et al., 2024b; Qiao et al., 2024b; Dong et al., 2024a) use programs to assist in data synthesis or training; for example, AutoIf (Dong et al., 2024a) filters synthetic training data through testing programs. Additionally, Lyu et al. (2024) explore the capability of LLMs to act as code interpreters to execute LeetCode programs. Distinguishing these studies, our work is unique in that it: (1) employs verification programs for the self-correction phase; (2) utilizes pseudo-programs that do not require strict executability; (3) enables LLMs to self-generate and self-execute programs without the necessity for actual symbolic tools.

Inference Framework. Unlike directly generating answers, many studies have explored enhancing LLMs' ability to solves complex tasks through reasoning frameworks. Decomposition-based methods like COT, TOT, and POT (Wei et al., 2022; Yao et al., 2024a; Chen et al., 2023; Qiao et al., 2024a) guide models to break down tasks step-by-step, while sampling-based methods (Wang et al., 2022; Zhang et al., 2024b; Dong et al., 2024c) explore diverse reasoning paths and select the most consistent or optimal response. Another key approach is the mechanism of reflection and self-correction (Zhang et al., 2024c; Shinn et al., 2024; Pan et al., 2024), which encourages models to iteratively evaluate and refine their responses. Additionally, to balance efficiency, some works (Yao et al., 2024b) dynamically combine fast and slow reasoning based on task complexity. ProgCo can be seen as a method of fast and slow reasoning, where RrogVe filters out complex tasks, and ProgRe uses dual optimization for slow reasoning. Moreover, since ProgCo focuses on the self-correction, it is orthogonal to forward reasoning methods like sampling and decomposition, and can be well combined with them.

B Details of Method

B.1 Details of Prompts

We present the prompts used for mathematical tasks in Figures 6, 7, 8, 9, 10, 11.

B.2 Pseudo-code

We summarize the pseudo-code of ProgCo in Algorithm 1.

C Details of Experiment

C.1 Details of Datasets

The introduction to the evaluation datasets is as follows:

- **IFEval** (Zhou et al., 2023): IFEval is one of the most commonly used instruction-following benchmark for LLMs, containing over 500 test samples and covering 25 types of atomic instructions. The Prompt(Strict) metric calculates the proportion of strict following of the input prompt across all samples. For a sample, the prompt is considered followed only if all atomic instructions in it are

Algorithm 1 : Program-driven Self-Correction (ProgCo)

Input: input x, model M, prompts $\{P^{gen}_{fuc}, P^{exec}_{fuc}, P_{fb}, P_{reflex}, P_{cont}, P^{code}_{reflex}\}$
Output: final response y_{final}

$y_0 = M(x)$ ▷ Initial generation
for iteration $i \in 0, 1, ..., I$ **do**
  $f_i = M(P^{gen}_{fuc}||x)$ ▷ Verification Program Generation (Eqn.1)
  $r_i = M(P^{exec}_{fuc}||x||y_i)$ ▷ Verification Program Execution (Eqn.2)
  $fb_i = M(P_{fb}||x||y_i||r_i)$
  if $fb_i =$ "pass" **then**
    break
  else
    $y^{temp}_{i+1} = M(P_{reflex}||x||y_i||fb_i)$ ▷ Preliminary Reflection (Eqn.3)
    if y^{temp}_{i+1} != y_i **then**
      $ins = M(P_{cont}||y_i||y^{temp}_{i+1})$ ▷ Contrast and Regenerate (Eqn.4)
      $y_{i+1} = M(ins||x)$
    else
      $y_{i+1} = y^{temp}_{i+1}$
    end if
    $f_{i+1} = M(P^{code}_{reflex}||x||f_i||y_i||fb_i)$ ▷ Verification Program Refinement (Eqn.5)
  end if
end for
$y_{final} = y_i$

followed. The Instruction(Strict) metric calculates the proportion of strict following of each atomic instruction across all samples.

- **GSM8K** (Cobbe et al., 2021): GSM8K is a classic mathematical reasoning evaluation dataset that primarily focuses on math problem-solving at the grade school level. The dataset contains over 8,000 samples, with 7,473 samples in the training set and 1,319 samples in the test set.

- **MATH** (Hendrycks et al., 2021): The MATH dataset consists of 12,500 complex math competition problems covering various branches of mathematics, including algebra, geometry, probability, and number theory. The MATH training set contains 7,500 samples, and the test set includes 5,000 samples. We randomly select 500 from the test set for evaluation.

C.2 Details of Baselines

The introduction to the baselines is as follows:

- **Vanilla-Reflex**: Vanilla-Reflex is a simple setup that iteratively prompts LLMs to reflect on its original output and generate a new output without an early stopping mechanism.

- **Self-Refine** (Madaan et al., 2024): Self-refine iteratively examines its own output to obtain feedback and suggests refining its output based on this feedback until the examination indicates that there are no errors.

- **Self-Reflection** (Shinn et al., 2024): For erroneous outputs from environmental feedback, Reflection will first reflect on the output and generate task experiences. These experiences will be incorporated into the next attempt as a way of refinement. In the absence of environmental feedback, we use COT checks consistent with Self-Refine.

- **CheckList** (Cook et al., 2024): CheckList first generates a verification list for the input task and then verifies whether the response satisfies all the verification points one by one. Any unsatisfied verification points will be used as feedback to optimize its own output.

C.3 Details of Implementation

We use azure GPT-3.5-Turbo-0613 (16K) for GPT-3.5 and GPT-4o-0806 for GPT-4o (OpenAI, 2024). For Llama3.1-8B-Instruct (Meta, 2024), we employ FastTransformer[3] to accelerate inference. The

[3]https://github.com/NVIDIA/FasterTransformer

inference temperature is set to 0, while other parameters are kept at their default values. We utilize few-shot demonstrations for both the baselines and ProgCo, adjusting the number of demonstrations to 1-3 based on the demonstration content length.

For instruction following, we use the official IFEval (Zhou et al., 2023) evaluation with Strict Prompt and Strict Instruction metrics. To achieve more accurate mathematical evaluation, we prompt GPT-4o-mini (OpenAI, 2024) to extract LLM's output answers and perform regex matching with the standard answers. Answers deemed incorrect by regex are further evaluated by GPT-4o-mini to determine their correctness, preventing errors in synonymous expressions, such as decimals and fractions.

Generate Verification Program

```
You are an expert in reverse reasoning verification. Given a problem, you need to generate a reverse
verification executable Python function for that problem.

[Reverse Reasoning Introduction]
* Reverse reasoning is a method of thinking that starts from the result and verifies the problem backwards
. Specifically, it involves:

1. Starting with the given answer, rather than the initial conditions of the problem.
2. Assuming this answer is correct, then conducting reverse deduction based on this assumption and the
known conditions in the problem.
3. Through this reverse deduction, checking whether other known conditions or constraints in the problem
can be satisfied.
4. If the results of the reverse deduction can satisfy all conditions, then the original answer can be
considered correct.

For example, using the answer and known conditions 1 and 2 as assumptions, reverse reason to check if it
satisfies known condition 3 in the problem statement.

To illustrate, consider a problem of solving a quadratic equation ax^2+bx+c=0:
- Forward thinking would start with a, b, c values and use the quadratic formula to calculate x.
- Reverse reasoning would:
1. Start with a possible solution x
2. Substitute it into ax^2 + bx + c
3. Check if the result equals 0
4. If it equals 0, the solution is verified as cor

[Requirements]
1. The verification function should start with the input answer and use reverse reasoning to validate the
correctness of the answer.
2. The verification function should only accept one input (the answer) and output the verification result
as True/False.
3. After generating the verification function name, please first write the reverse analysis verification
approach as code comments, then generate the content of the verification function. Please do not output
any other content.
```

Figure 6: The prompt for generating verification programs.

Execute Verification Program

```
You are a verification expert proficient in code execution and possessing extensive world knowledge, and
you have the following advantages:
1. Ability to execute code flexibly, unrestricted by strict syntax or standards
2. Capacity to comprehend code purpose, overall logic, and potential issues through analysis of comments
and context

Your task is to verify a given mathematical problem and its solution. A data annotator has written
verification code for this problem. Please proceed as follows:

1. Carefully read the provided problem, solution process, and answer
2. Execute the verification code step by step
3. If you discover any issues in the verification code, make appropriate revisions before continuing
execution
4. Derive the final verification result

Input information:
[Problem]
{query}
[Solution Process]
{response}
[Solution Answer]
{result}
[Verification Code]
{validate_response_fuc}

Please output your analysis and results strictly in the following format, without adding any additional
content:

[Execution of Verification Code]
{Detailed step-by-step execution process}
[Verification Result]
{True or False}
```

Figure 7: The prompt for executing verification programs.

Get Feedback

```
Extract key information from the execution process of the verification code and convert it into natural
language form.
[Problem]
{query}
[Execution of Verification Code]
{execute_content}
```

Figure 8: The prompt used to get feedback from execution.

Preliminary Reflection of Response

```
Your task is to reflex whether a solution is correct.

Given a problem [Problem], a reverse reasoning validation expert generated an executable Python function
for reverse validation [Initial Verification Code] of this problem.
However, the solution has not been validated by the verification function. This means that either the
solution or the verification function, or both, contain errors.

You need to carefully analyze and complete the following tasks:
1. Reflect on the initial solution process:
   - Compare in detail the solution and the feedback, examine the logic and accuracy of the solution
   approach **step by step**, consider whether the errors lie with the feedback or with the solution.
   - If errors or inadequacies are found, provide detailed feedback and suggestions for improvement
   - If it is found that there are no errors in the solution but rather errors exist in the feedback,
   state that the solution is correct.
2. Provide a new solution:
   - If the initial solution is correct, please use the original solution process (Note: If the
   verification code is correct, you also need to repeat the initial verification code word for word)
   - If errors or inadequacies are found, revise the solution based on your reflection.

Note:
- You cannot refuse to generate a new solution due to missing information or other reasons.
Please strictly output your analysis and revision according to the following format, without any
additional content:

[Reflection]
{Detailed reflection process}
[New Solution]
{Complete solution process based on the reflection}
```

Figure 9: The prompt for reflecting on responses.

Contrast on pre and post-reflection responses

```
You are an expert at comparing and extracting key points.
Task: Analyze the differences between two solutions and extract key points
Background: For the same problem, two solutions have provided different answers. We need to analyze these
differences in depth to identify the key aspects of the problem.

Steps:
1. Carefully read the problem and both solutions
2. Ignore surface differences in expression, focus on substantial differences in content and method
3. Compare the core ideas, key steps, and final results of both solutions
4. Summarize the essence of the problem reflected by these differences and the key points to note when
solving

Output requirements:
1. Concisely list 1-3 key points
2. Each point should be specific and helpful for regenerating a better solution

Output format:
[Comparative Analysis Process]
{Your comparative analysis process}
[Core Differences in Solutions]
{Summarize the differences in solutions based on the comparative analysis process, answer in bullet points
}
[Key Points to Note When Solving the Problem]
{Summarize the key points to note when solving the problem based on the differences in solutions, answer
in bullet points}
```

Figure 10: The prompt for contrasting responses and generating insights.

Reflection on the verification program

```
Your task is to reflex whether a verification code is correct.

[Reverse Reasoning Introduction]
* Reverse reasoning is a method of thinking that starts from the result and verifies the problem backwards
. Specifically, it involves:

1. Starting with the given answer, rather than the initial conditions of the problem.
2. Assuming this answer is correct, then conducting reverse deduction based on this assumption and the
known conditions in the problem.
3. Through this reverse deduction, checking whether other known conditions or constraints in the problem
can be satisfied.
4. If the results of the reverse deduction can satisfy all conditions, then the original answer can be
considered correct.

For example, using the answer and known conditions 1 and 2 as assumptions, reverse reason to check if it
satisfies known condition 3 in the problem statement.

To illustrate, consider a problem of solving a quadratic equation ax^2 + bx + c = 0:
- Forward thinking would start with a, b, c values and use the quadratic formula to calculate x.
- Reverse reasoning would:
1. Start with a possible solution x
2. Substitute it into ax^2 + bx + c
3. Check if the result equals 0
4. If it equals 0, the solution is verified as cor

Given a problem [Problem], a reverse reasoning validation expert generated an executable Python function
for reverse validation [Initial Verification Code] of this problem. However, the verification code did not
 pass when verifying a solution [Solution] of the problem.

The error may come from the solution or from the verification code.
You need to carefully analyze and complete the following tasks:
1. Reflect on the initial verification code:
   - First, examine the logic and accuracy of the verification code step by step
   - If errors or inadequacies are found, provide detailed feedback and suggestions for improvement
2. Provide a new verification code:
   - If the initial verification code is correct, you can use the original verification code
   - If errors or inadequacies are found, revise the verification code based on your reflection
   - Keep detailed comments in the revised validation code

Please strictly output your analysis and revision according to the following format, without any
additional content:

[Reflection]
{Detailed reflection process}
[New Verification Code]
{Complete verification code based on the reflection}
```

Figure 11: The prompt for reflecting on verification programs.

ProgVe Case on IFEval (GPT-3.5)

Instruction: Write a joke about a startup that sells dog food in a song. Your entire response should be in English, and in all capital letters. Your answer must contain a title, wrapped in double angular brackets, i.e. <<title>>.

Response:
<<title>> "Barking Business"
Why did the startup that sells dog food decide to sing about their business?
Because they wanted to unleash their catchy jingle and make tails wag to the beat!

Verification Program:

```
def validate_response(response):
    errors = []
    # Check if the response is in all capital letters
    if response != response.upper():
        errors.append("Response is not entirely in capital letters")
    # Check if the response is in English
    if not is_english(response):
        errors.append("Response is not in English")
    # Check if the response contains a title wrapped in double angular brackets
    if not has_title(response):
        errors.append("Response does not contain a title wrapped in double angular brackets")
    # If errors exist, return False and error messages; otherwise return True
    if errors:
        return False, "\n".join(errors)
    return True, "No Error"

# Auxiliary functions:
# is_english(text): Check if the text is in English
# has_title(text): Check if the text contains a title wrapped in double angular brackets
```

Verification Program Execution:
Step 1: Initialize errors list
errors = []
Step 2: Check if the response is in all capital letters
If response != response.upper():
    errors.append("Response is not entirely in capital letters")
The response is not entirely in capital letters,so this error is added to the list.
Step 3: Check if the response is in English
Assuming is_english(response) returns True (as it appears to be in English), no error is added.
Step 4: Check if the response contains a title wrapped in double angular brackets
If not has_title(response):
    errors.append("Response does not contain a title wrapped in double angular brackets")
The response does not contain a title wrapped in double angular brackets, so this error is added to the list.
Step 5: Return result
errors list contains two items: "Response is not entirely in capital letters" and "Response does not contain a title wrapped in double angular brackets"
Since errors is not empty, the function returns False and the error messages.

So, the final result of validate_response(response) is:
<result>False, "Response is not entirely in capital letters\nResponse does not contain a title wrapped in double angular brackets"</result>

Feedback:
The initial response needs to be modified based on the following suggestions:
1. Ensure that the response is entirely in capital letters.
2. Include a title wrapped in double angular brackets.

Figure 12: A case of ProgVe self-verifying on IFEval.

ProgVe Case on GSM8K (GPT-3.5)

Question:

```
Marissa is hiking a 12-mile trail. She took 1 hour to walk the first 4 miles, then another hour to walk
the next two miles. If she wants her average speed to be 4 miles per hour, what speed (in miles per hour)
does she need to walk the remaining distance?
```

Response:

```
Marissa has already walked 4 + 2 = <<4+2=6>>6 miles.
She has 12 - 6 = <<12-6=6>>6 miles left to walk.
She has spent 1 + 1 = <<1+1=2>>2 hours walking so far.
To average 4 miles per hour, she needs to walk the remaining 6 miles in 6 / 4 = <<6/4=1.5>>1.5 hours.
Thus, she needs to walk the remaining distance at a speed of 6 / 1.5 = <<6/1.5=4>>4 miles per hour. Answer
: \boxed{4}.
```

Verification Program:

```
def verify_remaining_speed(remaining_speed):
    # Known conditions
    total_distance = 12
    first_distance = 4
    first_time = 1
    second_distance = 2
    second_time = 1
    target_average_speed = 4

    # Calculate the time taken to walk the remaining distance
    remaining_time = (total_distance - first_distance - second_distance) / remaining_speed

    # Calculate the total time taken to walk the entire trail
    total_time = first_time + second_time + remaining_time

    # Calculate the average speed based on the total time and total distance
    average_speed = total_distance / total_time

    # Check if the average speed is 4 miles per hour
    if abs(average_speed - target_average_speed) > 0.01:  # Allow for small rounding errors
        return False

    # All conditions are satisfied
    return True
```

Execution of Verification Program:

```
Calculate the time taken to walk the remaining distance:
remaining_time = (total_distance - first_distance - second_distance) / remaining_speed
remaining_time = (12 - 4 - 2) / 4
remaining_time = 6 / 4
remaining_time = 1.5

Calculate the total time taken to walk the entire trail:
total_time = first_time + second_time + remaining_time
total_time = 1 + 1 + 1.5
total_time = 3.5

Calculate the average speed based on the total time and total distance:
average_speed = total_distance / total_time
average_speed = 12 / 3.5
average_speed = 3.4285714285714284

Check if the average speed is 4 miles per hour:
abs(average_speed - target_average_speed) = abs(3.4285714285714284 - 4) = 0.5714285714285716
0.5714285714285716 > 0.01, so this check fails.

The condition in step 5 fails, therefore the function returns False.
```

Feedback:

```
We know that Marissa is hiking a 12-mile trail. She took 1 hour to walk the first 4 miles and another hour
 to walk the next 2 miles. Her target average speed is 4 miles per hour.
First, let's calculate the time taken to walk the remaining distance. The remaining distance is 12 miles -
 4 miles - 2 miles = 6 miles. If Marissa wants her average speed to be 4 miles per hour, the time taken to
 walk the remaining distance would be 6 miles / 4 miles per hour = 1.5 hours.
Next, let's calculate the total time taken to walk the entire trail. The total time is the sum of the time
 taken for the first part, the second part, and the remaining part. So, the total time is 1 hour + 1 hour
+ 1.5 hours = 3.5 hours.
Then, let's calculate the average speed based on the total time and total distance. The average speed is
the total distance divided by the total time. So, the average speed is 12 miles / 3.5 hours =
3.4285714285714284 miles per hour.
However, the target average speed is 4 miles per hour, which does not match the calculated average speed
of 3.4285714285714284 miles per hour. Therefore, the solution of 4 miles per hour for the remaining speed
fails our verification process because it leads to inconsistencies with the known facts.
```

Figure 13: A case of ProgVe self-verifying on GSM8K.

ProgRe Case on GSM8K following ProgVe (GPT-3.5)

Preliminary Reflection of Response:

```
[Reflection]
The initial solution correctly calculates the remaining distance to be 6 miles (12 miles - 4 miles - 2
miles) and determines that Marissa needs to walk the remaining 6 miles in 1.5 hours to achieve an average
speed of 4 miles per hour. However, the feedback points out that the calculated average speed of
3.4285714285714284 miles per hour does not match the target average speed of 4 miles per hour.
Upon reviewing the feedback, it is clear that the initial solution made an error in assuming that Marissa
needs to walk the remaining distance at a speed of 4 miles per hour. This assumption is incorrect because
the target average speed of 4 miles per hour refers to the entire trail, not just the remaining distance.
[New Solution]
To find the speed Marissa needs to walk the remaining distance, we can calculate the time it took her to
walk the first 6 miles and subtract it from the total time.
Marissa took 1 hour to walk the first 4 miles and another hour to walk the next 2 miles, so she took a
total of 1 + 1 = 2 hours to walk the first 6 miles.
The remaining distance is 12 miles - 6 miles = 6 miles.
To find the speed, we divide the remaining distance by the remaining time:
Speed = Remaining Distance / Remaining Time
Speed = 6 miles / 2 hours
Speed = 3 miles per hour
Therefore, Marissa needs to walk the remaining distance at a speed of 3 miles per hour. Answer: \boxed{3}.
```

Contrast Insights:

```
- Ensure accurate calculation of the remaining time by considering the total time spent and the desired
average speed.
- Double-check calculations to ensure that the remaining speed is correctly calculated by dividing the
remaining distance by the remaining time.
```

New Response:

```
To find the speed Marissa needs to walk the remaining distance, we first need to calculate the remaining
time.
Marissa has already walked 4 miles in 1 hour and another 2 miles in another hour, so she has already spent
 1 + 1 = 2 hours on the trail.
Since she wants her average speed to be 4 miles per hour, we can calculate the total time she should spend
 on the trail by dividing the total distance (12 miles) by the desired average speed (4 miles per hour).
Total time = Total distance / Average speed
Total time = 12 miles / 4 miles per hour
Total time = 3 hours
Since Marissa has already spent 2 hours on the trail, the remaining time is 3 hours - 2 hours = 1 hour.
To find the speed Marissa needs to walk the remaining distance, we divide the remaining distance (12 miles
 - 4 miles - 2 miles = 6 miles) by the remaining time (1 hour).
Remaining speed = Remaining distance / Remaining time
Remaining speed = 6 miles / 1 hour
Remaining speed = 6 miles per hour
Therefore, Marissa needs to walk the remaining distance at a speed of 6 miles per hour.
```

Verification Program Refinement:

```
[Reflection]
The initial verification code attempts to validate the solution by following a reverse reasoning approach.
 However, there is an error in the logic used to check if the average speed is 4 miles per hour.
Here is a detailed reflection of the code:
- The code properly calculates the time taken to walk the remaining distance as `(total_distance -
first_distance - second_distance) / remaining_speed`.
- It then calculates the total time taken to walk the entire trail as the sum of the first time, second
time, and remaining time.
- The code calculates the average speed based on the total time and total distance using the formula `
average_speed = total_distance / total_time`.
- However, the check for the average speed being 4 miles per hour is incorrect. It compares the calculated
 average speed with the target average speed using a tolerance of `0.01`. This is incorrect as there is no
 need for a tolerance in this case. The two values should be exactly equal for validation.
[New Verification Program]:
def verify_remaining_speed(remaining_speed):
    # Known conditions
    total_distance = 12
    first_distance = 4
    first_time = 1
    second_distance = 2
    second_time = 1

    # Calculate the time taken to walk the remaining distance
    remaining_time = (total_distance - first_distance - second_distance) / remaining_speed

    # Calculate the total time taken to walk the entire trail
    total_time = first_time + second_time + remaining_time

    # Calculate the average speed based on the total time and total distance
    average_speed = total_distance / total_time

    # Check if the average speed is 4 miles per hour
    if average_speed != 4:
        return False

    # All conditions are satisfied
    return True
```

Figure 14: A case of ProgRe self-refining on GSM8K (following ProgVe).

Leveraging Self-Attention for Input-Dependent Soft Prompting in LLMs

Ananth Muppidi*
IIIT Hyderabad
India
ananth.muppidi21@gmail.com

Abhilash Nandy*
IIT Kharagpur
India
nandyabhilash@gmail.com

Sambaran Bandyopadhyay
Adobe Research
India
samb.bandyo@gmail.com

Abstract

The performance of large language models in domain-specific tasks necessitates fine-tuning, which is computationally expensive and technically challenging. This paper focuses on parameter-efficient fine-tuning using soft prompting, a promising approach that adapts pre-trained models to downstream tasks by learning a small set of parameters. We propose a novel Input Dependent Soft Prompting technique with a self-Attention Mechanism (*ID-SPAM*) that generates soft prompts based on the input tokens and attends different tokens with varying importance. Our method is simple and efficient, keeping the number of trainable parameters small. We show the merits of the proposed approach compared to state-of-the-art techniques on various tasks and show the improved zero shot domain transfer capability.

1 Introduction

Large language models (LLMs) have made significant advancements in natural language processing tasks, such as generation, translation and summarization (Yeo et al., 2023; Zhang et al., 2023a). Despite their success, LLMs' performance in domain-specific tasks is limited, and fine-tuning on task-oriented datasets is crucial. As models from BERT (Devlin et al., 2019) to GPT-3 (Brown et al., 2020) have millions to billions of parameters, fine-tuning becomes computationally expensive and challenging. Therefore, parameter efficient fine-tuning (Han et al., 2024) research aims to adapt pre-trained models to downstream tasks by fixing most parameters and only learning a small subset.

Soft prompting is a promising direction for fine-tuning large models. Without changing the core architecture of an LLM, soft prompt methods generally introduce a small trainable vector (known as a 'soft prompt') at the beginning of one or more transformer layers' inputs within the LLM. During fine tuning, only the soft prompt is trained to adapt to the downstream task keeping the parameters of the base LLM frozen. Lester et al. (2021) propose Prompt Tuning by prepending the trainable soft prompt vector before the embeddings of the text input, just after the embedding layer of the base LLM. On similar lines, Li and Liang (2021) introduce Prefix Tuning by prepending a soft prompt at every transformer layer and Liu et al. (2021) come up with P-tuning by interleaving learnable prompts with input embeddings. Contrary to text prompt engineering (Wei et al., 2022) or optimizing discrete token representations via in-context learning (Dai et al., 2023), Petrov et al. (2023) suggest that the continuous embedding space of soft prompts inherently possesses a greater amount of information.

Recent literature introduces several variants of soft prompt techniques such as removing the reparameterization module (Liu et al., 2022b), hierarchical structured pruning (Ma et al., 2022), introducing an adaptive gate mechanism to control the prefix importance in each transformer layer (Zhang et al., 2023b), diving the soft prompt into query, key and value prompts (Wang et al., 2023), learning multiple short soft prompts and a gating mechanism to route an input to a specific soft prompt (Choi et al., 2023), and decomposing the soft prompt into low rank matrices (Shi and Lipani, 2024).

Many of these methods keep the soft prompt independent of the actual input given to the LLM. However, this limits the soft prompt to adjust based on the actual input during the inference time. It is unlikely that a unified prompt would lead to a performance improvement across different input instances. It also makes the training difficult by increasing the convergence time. To address this, a few recent approaches leverage input dependent soft prompts. But they need to concatenate the soft prompts either at every transformer layer of the base LLM (Wu et al., 2022) or all the layers after an

*Equal contribution. Work done during the internship at Adobe Research.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 960–969
July 27 - August 1, 2025

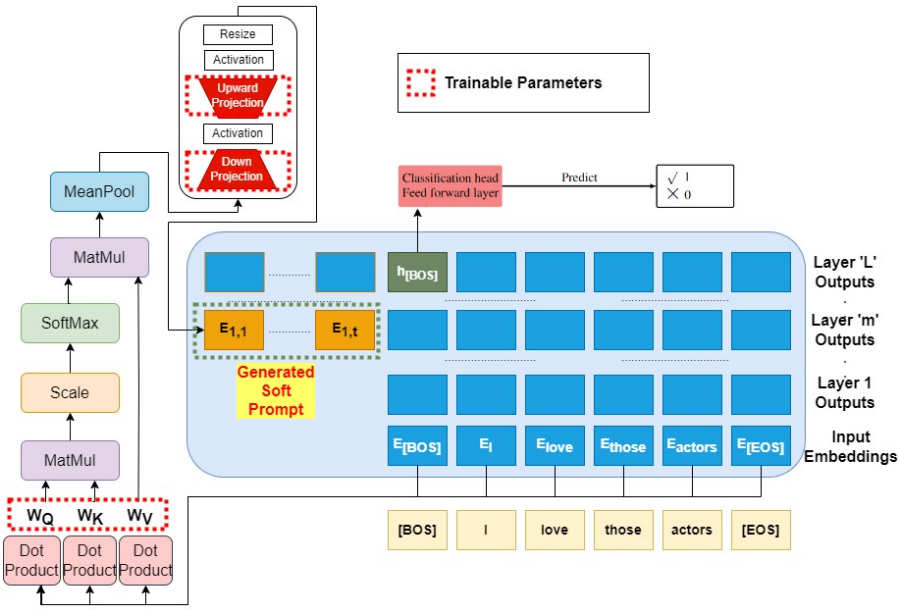


Figure 1: *ID-SPAM* Framework. Given an LM, the generated soft-prompt can be prepended to any transformer layer's inputs (the figure can be best seen in color)

intermediate layer (Liu et al., 2022a), or transform the soft prompt by using cross-attention with the input tokens without explicitly generating from them (Jin et al., 2023). These input dependent prompting techniques still have multiple limitations: (i) Many of them employ relatively complicated architecture by concatenating soft prompts in multiple internal transformer layers of the LLM; (ii) Since, a task may contain diverse samples with different types of words, it is important to attend different words of the input with different weights while generating the soft prompt; And (iii) Number of trainable parameters often increases significantly.

To address the above research gaps, we introduce an input dependent soft prompt technique where the soft prompt is generated by a trainable network that attends different tokens of the input with different importance by employing a self-attention mechanism. We prepend the soft prompt with the input to a single transformer layer of the base LLM, keeping the number of trainable parameters small and training smooth. Following are the *contributions* made in this work: **(i)** We propose *ID-SPAM*, a novel (Input Dependent Soft Prompting technique with a self-Attention Mechanism); Our method is simple and efficient to train. **(ii)** We show the merit of the proposed approach on six tasks from the GLUE benchmark (Wang et al., 2018); And **(iii)** Due to the use of trainable attention on the input tokens, our approach is more efficient in zero-shot domain transfer as shown in the experiment.

2 Proposed Solution

In this section, we introduce our proposed method *ID-SPAM* (see its framework in Figure 1).

Given a Task T having training data represented as $D_{train} = \{(x_i, y_i)\}_{i=1}^{K}$. Following Lester et al. (2021), we represent the input as $x_i = \mathbf{E}(\texttt{[SEP]}\, S_1\, \texttt{[SEP]}\, S_2\, \texttt{[EOS]})$ for a task with a pair of sentences S_1, S_2 as the input or $x_i = \mathbf{E}(\texttt{[SEP]}\, S_1\, \texttt{[EOS]})$ for for a task with a single sentence S_1 as the input, where $\mathbf{E}(\cdot)$ is the token embedding for the input sentence(s).

We introduce a learnable soft prompt such that the prompt not only varies with the task at hand, but is also generated based on the input in such a way that it primarily attends to those input tokens that are essential for the given task. To make the learning efficient, we freeze the parameters of the original LM M. Our proposed soft prompt for the task T can be defined as $\mathbf{S}_T \in \mathbb{R}^{n \times t}$, where t is the number of tokens in the prompt representation and n is the hidden dimension of the LM $\mathbf{M}$ under consideration. $\mathbf{S}_T$ is obtained by first applying a *learnable* attention layer (Vaswani et al., 2017) over the input embeddings $\mathbf{E}(\cdot)$ and averaging the outputs, providing a context-rich representation. The $n \times 1$ dimensional vector A so obtained is passed through a downward projection MLP Layer having learnable weights $\mathbf{W}_{down} \in \mathbb{R}^{n \times c}$ and bias $\mathbf{b}_{down} \in \mathbb{R}^{c}$, followed by a ReLU Activation Layer (Nair and Hinton, 2010), and then an upward projection MLP

Layer having learnable weights $\mathbf{W}_{up} \in \mathbb{R}^{c \times n.t}$ and bias $\mathbf{b}_{down} \in \mathbb{R}^{n.t}$, where $c < n$. The output so obtained is re-sized to get the learnable, input-dependent soft prompt $\mathbf{S}_T \in \mathbb{R}^{n \times t}$, which is either prepended to the token embeddings or to the input of any intermediate transformer layer of the LM $\mathbf{M}$. We will show some analysis on the choice of intermediate layer in the experiments. Mathematically,

$$A = \text{mean}\left\{\text{softmax}\left(\frac{(\mathbf{EW}_Q)(\mathbf{EW}_K)^\top}{\sqrt{d_k}}\right)(\mathbf{EW}_V)\right\} \tag{1}$$

$$\mathbf{S}_T = \text{resize}(\sigma(W_{up}\sigma(W_{down}(A)))) \tag{2}$$

W_Q, W_K, and W_V are the query, key, and value parameter matrices respectively, and $\frac{1}{\sqrt{d_k}}$ is a scaling factor, as used in Vaswani et al. (2017). σ is a non-linear activation which we used ReLU here.

3 Experimental Evaluation

Here, we describe our experimental setup, evaluate *ID-SPAM* framework on GLUE and SuperGLUE benchmarks, and zero-shot domain transfer between tasks against several baselines, followed by a detailed analysis.

3.1 Experimental Setup

We compare *ID-SPAM* with the following baselines - (1) **Transformer fine-tuning:** Here, all parameters of LM are learned (2) **Parameter-Efficient Soft Prompt-based Methods -** (a) **Prompt Tuning:** We use standard prompt tuning (Lester et al., 2021), which learns soft prompts through back-propagation to condition frozen language models for specific tasks. (b) **P-tuning:** P-tuning (Liu et al., 2022b) is a variant of Deep Prompt Tuning (Li and Liang, 2021; Qin and Eisner, 2021) adapted for NLU (c) **Sparse Mixture of Prompts (SMoP):** SMoP (Choi et al., 2023) leverages multiple short soft prompts with a gating mechanism to train multiple prompts tailored in addressing different data subsets (d) **Late Prompt Tuning (LPT):** LPT (Liu et al., 2022a) injects a late prompt into an intermediate layer of the LM, rather than into the input layer or across all layers. (e) **Decomposed Prompt Tuning (DEPT):** DEPT (Shi and Lipani, 2024) employs a decomposition strategy for the soft prompt, breaking it down into a pair of low-rank matrices. These components are then optimized independently, each with its own specific learning rate. (3) **Parameter Efficient Fine-tuning using Low-Rank Adaptation (LoRA):** LoRA (Hu et al., 2022) addresses challenge of fine-tuning large language models by freezing pre-trained model's weights and introducing trainable low-rank matrices into each layer. Note that it does not use a soft prompt.

For all methods, we train upto 30 epochs (***Section E of Appendix*** shows convergence after 30 epochs) using Standard Cross-Entropy Loss and Adam Optimizer (Loshchilov and Hutter, 2018), and number of soft-prompt tokens $t = 10$. We perform hyperparameter tuning for *ID-SPAM*, as described ***in Section A of Appendix***. We use a NVIDIA A100 GPU with a VRAM of 80 GB for all experiments.

3.2 Evaluation on GLUE Benchmark

We evaluate *ID-SPAM* and baselines on the following 6 Natural Language Understanding (NLU) Tasks from GLUE Benchmark (Wang et al., 2018) - SST-2 (Socher et al., 2013), MRPC (Dolan and Brockett, 2005), MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), RTE (Dagan et al., 2005), and QQP (Quora, 2017). These tasks cover various aspects of natural language understanding and inference, providing a comprehensive assessment of our approach's performance across different language processing challenges. All datasets were obtained from the Hugging Face library (Wolf et al., 2020; Lhoest et al., 2021). Further dataset statistics are shared in Table 1.

We report accuracy for {SST, MNLI, QNLI, RTE} and average of accuracy and macro F1-Score for {MRPC, QQP} using RoBERTa-BASE, RoBERTa-LARGE backbones (Liu et al., 2019) in Table 2.

We infer that *ID-SPAM* outperforms all Parameter-Efficient Soft Prompt-based baselines on 4 out of 6 GLUE tasks and w.r.t average task performance, and is a close second for 2 tasks, when using both RoBERTa-BASE and RoBERTa-LARGE backbones. This could be attributed to the attention layer followed by 2-layer MLP in *ID-SPAM*, which efficiently generates a context-rich soft prompt. Also, *ID-SPAM* is shown to be more or similarly efficient compared to well-performing LPT baseline ***in Section D of Appendix***.

Section B of Appendix shows - *ID-SPAM* performs better than Soft Prompt baselines - (1) on 2/4 and 3/4 **SuperGLUE** (Wang et al., 2019) tasks using RoBERTA-BASE and RoBERTA-LARGE backbones respectively, while giving best average score; (2) when using autoregressive GPT-2 backbone on 3/6 and 2/4 GLUE and SuperGLUE tasks

Category	Datasets	|Train|	|Dev|	|Labels|	Type	Labels
Single-sentence	SST-2	67349	872	2	sentiment	positive, negative
Sentence-pair	MNLI	392702	19647	3	NLI	entailment, neutral, contradiction
	MRPC	3668	408	2	paraphrase	equivalent, not equivalent
	QNLI	104743	5463	2	NLI	entailment, not entailment
	QQP	363846	40430	2	paraphrase	equivalent, not equivalent
	RTE	2490	277	2	NLI	entailment, not entailment

Table 1: Statistics of the datasets used in our experiments.

	MNLI	QNLI	SST-2	MRPC	RTE	QQP	Mean
Method	**GLUE (RoBERTa-BASE Backbone)**						
Fine-tuning	$87.4_{2.4}$	$91.3_{1.0}$	$92.3_{0.6}$	$92.7_{0.7}$	$82.5_{1.3}$	$90.9_{0.8}$	89.5
LoRA	$88.7_{0.4}$	$84.2_{2.1}$	$90.4_{0.3}$	$79.3_{0.5}$	$77.6_{1.1}$	$81.8_{0.2}$	83.7
Prompt Tuning	$78.3_{2.1}$	$81.4_{1.1}$	$89.3_{1.4}$	$74.4_{0.7}$	$57.9_{0.5}$	$77.8_{1.6}$	76.5
P-Tuning	$82_{2.2}$	$82.5_{0.3}$	$88.1_{0.5}$	$81.9_{1.7}$	$67.4_{0.9}$	$84.2_{0.1}$	81
SMoP	$80.7_{1.0}$	$82.9_{1.4}$	$89.8_{0.3}$	$78.1_{2.1}$	$71.7_{1.8}$	$83.7_{0.9}$	81.2
LPT	$81.7_{0.6}$	$83.2_{1.1}$	$91.8_{1.3}$	$\mathbf{84.3}_{0.2}$	$73.6_{0.7}$	$84.1_{0.5}$	83.1
DEPT	$81.5_{0.3}$	$\mathbf{87.9}_{1.2}$	$90.2_{1.2}$	$75.7_{0.6}$	$71.2_{1.0}$	$79.2_{0.3}$	81.0
ID-SPAM (ours)	$\mathbf{83.1}_{0.8}$	$86.4_{0.4}$	$\mathbf{92.7}_{1.2}$	$82.8_{0.3}$	$\mathbf{79.2}_{0.4}$	$\mathbf{84.6}_{0.5}$	**84.8**
Method	**GLUE (RoBERTa-LARGE Backbone)**						
Fine-tuning	$87.6_{1.7}$	$94.7_{2.3}$	$95.4_{1.3}$	$92.1_{1.2}$	$88.4_{0.3}$	$90.7_{0.2}$	91.48
LoRA	$89.1_{1.1}$	$87.9_{0.3}$	$95.1_{0.2}$	$86.5_{0.9}$	$78.7_{0.1}$	$88.4_{0.3}$	87.6
Prompt Tuning	$83.4_{1.1}$	$88.2_{0.2}$	$92.6_{0.5}$	$73.9_{1.4}$	$60.8_{0.6}$	$81.2_{0.6}$	80.0
P-Tuning	$86.4_{0.7}$	$88.7_{1.2}$	$\mathbf{95.8}_{0.8}$	$76.3_{1.1}$	$62.6_{0.5}$	$85.2_{1.3}$	82.5
SMoP	$86.7_{1.1}$	$88.4_{2.2}$	$\mathbf{95.8}_{1.4}$	$79.6_{0.8}$	$76.3_{1.4}$	$86.7_{0.3}$	85.6
LPT	$84.2_{1.1}$	$86.1_{0.5}$	$93.4_{1.4}$	$\mathbf{87.3}_{0.2}$	$74.2_{0.7}$	$85.3_{1.3}$	85.1
DEPT	$83.3_{1.2}$	$88.8_{1.3}$	$91.2_{1.8}$	$77.7_{0.3}$	$73.2_{0.8}$	$82.2_{0.7}$	82.7
ID-SPAM (ours)	$\mathbf{87.4}_{0.8}$	$\mathbf{91.1}_{0.4}$	$94.6_{1.2}$	$86.1_{0.3}$	$\mathbf{81.1}_{0.4}$	$\mathbf{88.4}_{0.5}$	**88.1**

Table 2: Test results on GLUE benchmark. We use RoBERTa-BASE, RoBERTa-LARGE Backbones for all methods. We report the score, along with stddev for 3 runs (in the subscript) for all tasks. The best performing Soft Prompt-based method's results are in bold

respectively, while giving better average score; (3) on average when using a GPT-2 Large Backbone.

Comparison with LoRA: *ID-SPAM* gives better average score compared to LoRA. Specifically, *ID-SPAM* outperforms LoRA in 5/6 and 3/6 tasks when using RoBERTa-BASE and RoBERTa-LARGE backbones respectively. Also, *ID-SPAM* is shown to be more efficient than LoRA based on the number of trainable parameters and training and inference times ***in Section D of Appendix***.

Ablation Analysis: We compare the results of *ID-SPAM* with just using mean-pooling directly using the RoBERTa-LARGE backbone on 3 GLUE Datasets in Table 3. *ID-SPAM* outperforms mean-pooling on all 3 tasks, giving an average improvement of 5.82%, thus highlighting the importance of the self-attention layer in *ID-SPAM*.

Method	MRPC	RTE	QQP
Mean-pooling	82.3	75.2	84.2
ID-SPAM	**86.1**	**81.1**	**88.4**

Table 3: Ablation Analysis on *ID-SPAM*

3.3 Evaluation on SuperGLUE Benchmark

We compare *ID-SPAM* with several Soft Prompt-Based Baselines on 4 SuperGLUE Datasets using RoBERTA-BASE and RoBERTA-LARGE backbones in Tables B and 5 respectively. We observe that *ID-SPAM* outperforms the baselines on 2/4 and 3/4 tasks using RoBERTA-BASE and RoBERTA-LARGE backbones respectively, while also giving the best average score.

	CB	COPA	MultiRC	BoolQ	Mean
Prompt Tuning	75.9	52.5	67.2	63.6	64.8
P-Tuning	76.3	54.7	67.9	63.7	65.6
SMoP	79.9	57.7	67.2	69.7	68.6
LPT	80.6	**59.2**	70.8	66.3	69.2
DEPT	78.6	52.9	67.1	**71.4**	67.5
ID-SPAM	**83.9**	57.8	**72.9**	69.9	**71.1**

Table 4: Test results on 4 SuperGLUE Datasets using RoBERTa-BASE Backbone. The best performing method is bold for each task.

	CB	COPA	MultiRC	BoolQ	Mean
Prompt Tuning	78	53	67.2	63.3	65.4
P-Tuning	76	55	68.1	64.0	65.8
SMoP	81.9	59	69.6	**71.1**	70.4
LPT	82	**60**	71.0	68.0	70.2
DEPT	79	54	69.0	71.0	68.2
ID-SPAM	**85**	**60**	**73.0**	70.0	**72.0**

Table 5: Test results on 4 SuperGLUE Datasets using RoBERTa-LARGE Backbone. The best performing method is bold for each task.

3.4 Zero-Shot Task, Domain Transfer

Table 6 shows Zero-Shot Transfer using RoBERTa-LARGE backbone, where a model is trained on training set of a dataset, evaluated on another dataset. We use (QQP, MRPC) and (SST-2, IMDB)[1] pairs for transfer across tasks and domains respectively similar to Lester et al. (2021). Table 6 shows *ID-SPAM* performs better than Soft Prompt-based baselines, showing *ID-SPAM* is generaliz-

[1]Task for SST-2 and IMDB is binary classification. SST-2 contains phrases, while IMDB contains full movie reviews

able across datasets. *ID-SPAM* even outperforms Fine-tuning in 3/4 pairs. Also, even though *ID-SPAM* has much less number of parameters compared to LoRA (***see Section D of Appendix***), *ID-SPAM* gives better/comparable performance. In addition, we show that *ID-SPAM* performs better/comparable to well-performing LPT baseline in Few-Shot Task Transfer ***in Section C of Appendix.***

Tuning Method	QQP→ MRPC	MRPC→ QQP	SST-2→ IMDB	IMDB→ SST-2
Fine-tuning	$64.0_{0.7}$	$68.3_{1.3}$	$87.1_{0.0}$	$88.8_{0.4}$
LoRA	$71.1_{0.1}$	$66.1_{0.4}$	$90.3_{0.2}$	$87.6_{1.1}$
Prompt Tuning	$54.1_{0.3}$	$54.6_{0.2}$	$68.7_{1.1}$	$63.5_{3.8}$
P-Tuning	$57.6_{1.2}$	$52.7_{1.1}$	$66.5_{0.0}$	$66.8_{1.3}$
SMoP	$67.9_{0.4}$	$64.1_{0.6}$	$84.5_{0.5}$	$83.3_{1.0}$
LPT	$66.7_{0.4}$	$64.5_{0.3}$	$67.1_{0.8}$	$71.1_{1.6}$
DEPT	$63.3_{1.8}$	$58.8_{0.5}$	$69.8_{0.1}$	$69.3_{0.9}$
ID-SPAM (ours)	$\mathbf{70.9}_{1.2}$	$\mathbf{69.2}_{0.4}$	$\mathbf{89.1}_{0.3}$	$\mathbf{86.0}_{0.8}$

Table 6: Mean, stddev of zero-shot task, domain transfer for different methods. 'Score' is average of Accuracy and macro F1-Score. The best performing Soft Prompt-based method's results are in bold.

3.5 Method Analysis

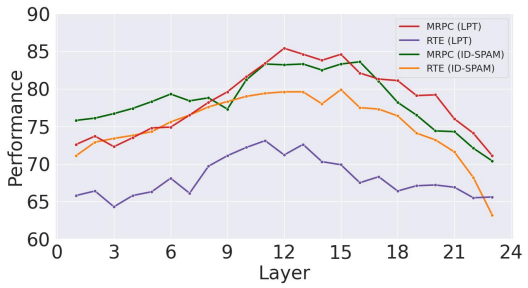


Figure 2: Effect of Variation in layer index (m) corresponding to which soft prompt is prepended on performance ($m = 0$ refers to input embeddings). Metrics are average of acc. and F1 for MRPC and acc. for RTE.

We analyze the effect of varying layer index where soft prompt is prepended (m in Figure 1) on performance of LPT and *ID-SPAM* on 2 GLUE datasets using RoBERTa-LARGE backbone in Figure 2. We infer that *ID-SPAM* and LPT perform better when soft prompt is prepended to inputs in middle layers of LM. Also, *ID-SPAM* significantly outperforms LPT corresponding to almost every layer index for RTE Dataset. Also, *ID-SPAM* performs better for earlier layers, as soft prompt is generated by using a single attention layer over input embeddings. Hence, prepending this prompt to an earlier layer's outputs performs better than later layer's outputs, as later layer's outputs are obtained after input embeddings are passed through several attention layers, reducing compatibility with the soft prompt. Also, if we prepend soft prompt to later layers, it passes through a small number of layers of LLM, thus showing a reduced performance.

4 Discussions and Conclusion

In this paper, we propose *ID-SPAM* which significantly improves parameter-efficient fine-tuning and zero-shot task and domain transfer performance on various NLU tasks compared to several SOTA parameter-efficient baselines. Notably, further analysis shows that *ID-SPAM* performs reasonably well when the generated soft prompt is prepended at any layer's inputs. Hence, *ID-SPAM* is an efficient, input-dependent soft prompt generation framework that could generalize well across several NLP tasks.

5 Limitations

We have shown that our proposed approach *ID-SPAM* improves the performance of two backbone LLMs (RoBERTa-BASE and RoBERTa-LARGE) on multiple NLP tasks. Our framework is generic and can be used with any open source LLMs as backbone. However, we could not use more recent very large scale pre-trained LLMs (like Llama-3.1-70B and Mixtral 8x22B) with tens of billions of parameters as backbone LMs in our experiments due to limited computational resources. We are interested to see the performance gain when we use our approach with those large scale state-of-the-art LLMs in some future work.

In the current work, we do not have an automated way to choose the layer of the LM where we input the soft prompt. The layer number is kept as a hyperparameter in the current work and its effect is shown in Section 3.5. In future, we want to automatically identify the optimal transformer layer, as proposed by Zhu and Tan (2023).

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Joon-Young Choi, Junho Kim, Jun-Hyung Park, Wing-Lam Mok, and SangKeun Lee. 2023. Smop: Towards efficient and effective prompt tuning with sparse mixture-of-prompts. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, volume 1.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP)*.

Zeyu Han, Chao Gao, Jinyang Liu, Sai Qian Zhang, et al. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Feihu Jin, Jinliang Lu, Jiajun Zhang, and Chengqing Zong. 2023. Instance-aware prompt learning for language understanding and generation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(7):1–18.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Quentin Lhoest, Albert Villanova del Moral, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Timo Müller, Isabella Géron, Simon Brandeis, Sylvain Gugger, Théo Matussière, Abhishek Thakur, Philipp Schmid, Yacine Jernite, Jeff Boudier, Francesco Calefato, Clara Ma, Clement Delangue, Thibault Goehringer, Victor Sanh, Canwen Xu, Alexander M. Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. *Preprint*, arXiv:2109.02846.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Xiangyang Liu, Tianxiang Sun, Xuan-Jing Huang, and Xipeng Qiu. 2022a. Late prompt tuning: A late prompt could be better than many prompts. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1325–1338.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022b. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv:2103.10385*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Fang Ma, Chen Zhang, Lei Ren, Jingang Wang, Qifan Wang, Wei Wu, Xiaojun Quan, and Dawei Song. 2022. Xprompt: Exploring the extreme of prompt tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11033–11047.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA. Omnipress.

Aleksandar Petrov, Philip Torr, and Adel Bibi. 2023. When do prompting and prefix-tuning work? a theory of capabilities and limitations. In *The Twelfth International Conference on Learning Representations*.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

Quora. 2017. Quora question pairs.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Zhengxiang Shi and Aldo Lipani. 2024. DePT: Decomposed prompt tuning for parameter-efficient fine-tuning. In *The Twelfth International Conference on Learning Representations*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Qifan Wang, Yuning Mao, Jingang Wang, Hanchao Yu, Shaoliang Nie, Sinong Wang, Fuli Feng, Lifu Huang, Xiaojun Quan, Zenglin Xu, et al. 2023. Aprompt: Attention prompt tuning for efficient adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9147–9160.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Rui Hou, Yuxiao Dong, VG Vinod Vydiswaran, and Hao Ma. 2022. Idpg: An instance-dependent prompt generation method. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5507–5521.

Yee Hui Yeo, Jamil S Samaan, Wee Han Ng, Peng-Sheng Ting, Hirsh Trivedi, Aarshi Vipani, Walid Ayoub, Ju Dong Yang, Omer Liran, Brennan Spiegel, et al. 2023. Assessing the performance of chatgpt in answering questions regarding cirrhosis and hepatocellular carcinoma. *medRxiv*, pages 2023–02.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023a. Summit: Iterative text summarization via chatgpt. *arXiv preprint arXiv:2305.14835*.

Zhen-Ru Zhang, Chuanqi Tan, Haiyang Xu, Chengyu Wang, Jun Huang, and Songfang Huang. 2023b. Towards adaptive prefix tuning for parameter-efficient language model fine-tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1239–1248.

Wei Zhu and Ming Tan. 2023. Spt: Learning to selectively insert prompts for better prompt tuning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Appendix

A Experiment Settings

For our experiments, we use `roberta-base` and `roberta-large` implementations from HuggingFace. For all baselines, the number of appended prompt tokens (for Prompt Tuning, P-tuning, Late Prompt Tuning) are set to 10 tokens. For DEPT, we set the rank to 45. For P-Tuning, we set the encoder reparameterization type to MLP. For *ID-SPAM*, appended prompt tokens are set to 10 tokens. The search space for hyperparameters for tuning are shown in Table 7. For all experiments, standard CrossEntropyLoss was used. For all experiments, we train using a warm-up rate of 0.06, and AdamW optimizer with ϵ of 1×10^{-6}, β_1 of 0.9, β_2 of 0.98.

In Figure 2, we can see that layers 11-13 show optimal performance for both *ID-SPAM* and LPT. LPT (Liu et al., 2022a) shows that the 13th layer is optimal. This makes our method *ID-SPAM* comparable to LPT taking the layer number into account. Also, following the trend from other prior art on soft prompts (Lester et al., 2021; Liu et al., 2022a; Li and Liang, 2021; Choi et al., 2023), we used the best hyperparameter set for each of the baselines. Our experimental approach is also logical and consistent as the experimental settings (choice of backbone LMs, datasets) are same for baselines and our method *ID-SPAM*.

Hyperparameter	Values
Epochs	{1, 5, 10, 20, 30}
Batch Size	{16, 32, 64}
Learning Rates	{1e-3, 5e-4, 1e-4, 5e-3, 1e-5}
Dropout Rate	{0.1, 0.2, 0.3}
Weight Decay	{0, 0.01, 0.1}
Layer (RoBERTa-Large)	{1,2,3...23}
Layer (RoBERTa-Base)	{1,2,3...11}

Table 7: Hyperparameters used for tuning *ID-SPAM*.

B Evaluation using GPT-2 and GPT-2 Large Backbones

Using GPT-2 Backbone. We carry out experiments with decoder-only GPT-2 backbone on 6 GLUE Datasets - Table 8 shows that when using GPT-2 as backbone, *ID-SPAM* outperforms LPT on 3/6 tasks and gives an average performance improvement of 2.3%.

Next, we carry out experiments with decoder-only GPT-2 backbone on 4 SuperGLUE Datasets –

	MNLI	QNLI	SST-2	RTE	QQP	MRPC	AVG
LPT	69.5	**79.4**	**90.1**	62.8	80.3	**81.9**	77.3
ID-SPAM	**78.3**	77.1	85.1	**71.6**	**82.9**	79.5	**79.1**

Table 8: Test results on 6 GLUE Datasets using GPT-2 Backbone. The best performing PEFT method is bold for each task.

Table 9 shows that compared to Soft Prompt-Based baselines, *ID-SPAM* gives the best average score, and performs the best on 2 tasks, while performing the second best on one of them.

	CB	COPA	MultiRC	BoolQ	Mean
Prompt Tuning	71.7	57	61.7	64.1	63.6
P-Tuning	73.3	57.7	63.2	65.7	65
SMoP	81.4	61.2	68.4	69.4	70.1
LPT	82.1	61.3	72.1	**74.1**	72.4
DEPT	76.1	55.1	**73.5**	67.2	68
ID-SPAM	**88.1**	**63.1**	71.7	72.4	**73.8**

Table 9: Test results on 4 SuperGLUE Datasets using GPT-2 Backbone. The best performing method is bold for each task.

Using GPT-2 Large Backbone. We compare the performance of *ID-SPAM* with LoRA and LPT using a large generative model GPT-2 Large (around 0.8 Billion Parameters) as the backbone on 2 GLUE Datasets - RTE and MRPC, as shown in Table 10.

Method	RTE	MRPC	Average
LoRA	74.0	80.0	77.0
LPT	69.9	82.9	76.4
ID-SPAM	73.7	81.1	**77.4**

Table 10: Test results on 2 GLUE Datasets using GPT-2 Large Backbone.

ID-SPAM gives an average improvement of 0.5% and 1.3% compared to LoRA and LPT respectively across the 2 GLUE Datasets, showing that *ID-SPAM* is competitive even for a large, generative backbone LM.

C Few-Shot Task Transfer

Train	Eval (Few-shot, 100 samples)	Tuning	Score
MRPC	QQP	Fine-Tuning	81.7
MRPC	QQP	LPT	74.4
MRPC	QQP	*ID-SPAM*	73.1
QQP	MRPC	Fine-Tuning	79.7
QQP	MRPC	LPT	69.4
QQP	MRPC	*ID-SPAM*	72.5

Table 11: Few-shot task transfer for different methods using the RoBERTa-LARGE Backbone.

ID-SPAM and LPT (a well-performing baseline in Table 2) using the RoBERTa-LARGE Backbone are fine-tuned on the first dataset, and then further fine-tuned on 100 training samples from the second. This model is then evaluated on the second dataset.

From Table 11, we can see that *ID-SPAM* performs better than LPT on QQP->MRPC, while the performance is comparable for MRPC->QQP.

D Comparison of *ID-SPAM* with baselines w.r.t model size and training and inference times

Model	LPT	LoRA	*ID-SPAM*
RoBERTa-BASE	2,162,688	3,495,312	2,064,384
RoBERTa-LARGE	2,883,584	7,931,280	3,538,944

Table 12: number of trainable parameters of *ID-SPAM* and well-performing baselines LPT and LoRA (see Table 2).

Table 12 shows that the number of trainable parameters in *ID-SPAM* is lesser than that of LoRA for both backbones, and is lesser than that of LPT using RoBERTa-BASE backbone, while they are comparable in case of RoBERTa-LARGE backbone.

Backbone	No. of Parameters in Backbone LM	*ID-SPAM*	LoRA
GPT2	126.8	2.1	2.4 (1.1x)
GPT2-medium	361.1	3.5	6.3 (1.8x)
GPT2-large	785.8	5.1	11.8 (2.3x)
GPT2-xl	1577.3	8.3	19.7 (2.4x)
Gemma-2B (Team et al., 2024)	2525.8	13.4	19.6 (1.5x)
FLAN-T5-xl (Chung et al., 2024)	2823.6	13.4	35.5 (2.6x)

Table 13: Number of trainable parameters (in millions) of *ID-SPAM* compared to LoRA for several LM backbones of different sizes. The decrease in the number of trainable parameters of *ID-SPAM* compared to LoRA is written within brackets.

Table 13 shows that as the size of the backbone LM increases, efficiency in the number of trainable parameters of *ID-SPAM* compared to LoRA tends to increase. Hence, *ID-SPAM* is suitable even for massive LMs.

Dataset	Method	Training Time per sample (in secs)	Inference Time per sample (in secs)
BoolQ	LPT	0.669	0.236
BoolQ	LoRA	0.715	0.313
BoolQ	*ID-SPAM*	0.651	0.251
WiC	LPT	0.082	0.041
WiC	LoRA	0.113	0.067
WiC	*ID-SPAM*	0.084	0.035

Table 14: Training and inference times of ID-SPAM and well-performing baselines LPT and LoRA for 2 SuperGLUE Datasets.

Table 14 shows that *ID-SPAM* requires less time for training as well as for inference, in comparison to LoRA on both BoolQ (a yes/no QA dataset) and WiC (dataset for binary classification) Datasets (2 datasets from SuperGLUE). Also, *ID-SPAM* takes lesser time to train on BoolQ than LPT, while the times are comparable on WiC. In case of inference, *ID-SPAM* takes lesser time than LPT for WiC, while taking slightly more time than LPT for BoolQ. Hence, *ID-SPAM* has comparable training and inference times w.r.t LPT, while giving better performance on GLUE datasets (see Table 2).

	MNLI	QNLI	SST-2	RTE	QQP	MRPC
Fine Tuning	2887s	270s	224s	247s	1854s	87s
LPT	2013s	157s	161s	168s	1157s	59s
ID-SPAM	1902s	166s	171s	168s	1004s	51s

Table 15: Total training time cost before convergence (in seconds) of *ID-SPAM* compared to baselines

Table 15 shows the training convergence times (in seconds - lower the better) for LPT and our proposed *ID-SPAM* (By convergence, we mean the epoch where the validation error is the least) using RoBERTa-LARGE Backbone. We can see that *ID-SPAM* gives better/similar convergence time compared to LPT on 4 out of 6 GLUE Tasks. Also, LPT takes an average convergence of time of 619 s, while ID-SPAM takes 577 s, giving an improvement of 7.3% in average convergence time.

E Convergence of the LoRA Baseline

The training loss is tabulated every 5 epochs in Table 16 when training LoRA with the RoBERTa-BASE backbone on the MRPC and RTE Datasets from the GLUE Benchmark.

Epoch	MRPC	RTE
5	0.21	0.4
10	0.12	0.14
15	0.05	0.07
20	0.02	0.06
25	0.02	0.04
30	0.0001	0.02

Table 16: Training Loss across epochs when training LoRA with the RoBERTa-BASE backbone

We can see that the training loss continuously decreases with increasing epochs on both the MRPC and RTE Datasets. Also, the losses are considerably lowered after 30 epochs as can be seen in the table, thus showing convergence.

Inconsistent Tokenizations Cause Language Models to be Perplexed by Japanese Grammar

Andrew Gambardella, Takeshi Kojima, Yusuke Iwasawa, Yutaka Matsuo
The University of Tokyo
Correspondence: atgambardella@weblab.t.u-tokyo.ac.jp

Abstract

Typical methods for evaluating the performance of language models evaluate their ability to answer questions accurately. These evaluation metrics are acceptable for determining the extent to which language models can understand and reason about text in a general sense, but fail to capture nuanced capabilities, such as the ability of language models to recognize and obey rare grammar points, particularly in languages other than English. We measure the perplexity of language models when confronted with the "first person psych predicate restriction" grammar point in Japanese. Weblab is the only tested open source model in the 7-10B parameter range which consistently assigns higher perplexity to ungrammatical psych predicate sentences than grammatical ones. We give evidence that Weblab's uniformly bad tokenization is a possible root cause for its good performance, and show that Llama 3's perplexity on grammatical psych predicate sentences can be reduced by orders of magnitude (28x difference) by restricting test sentences to those with uniformly well-behaved tokenizations. We show in further experiments on machine translation tasks that language models will use alternative grammar patterns in order to produce grammatical sentences when tokenization issues prevent the most natural sentence from being output.

1 Introduction

It is unusually difficult to find the correct tests and metrics to evaluate large language model (LLM) performance. Standard benchmarks such as MMLU (Hendrycks et al., 2021) are designed to capture the memorization and reasoning abilities of LLMs, but as LLMs are general tools which can be used to solve a large variety of tasks, there are many aspects of LLM performance which these benchmarks cannot cover.

In particular, these benchmarks are generally English-centric, and tend to test "higher-level" abilities, whereas more nuanced evaluation of LLMs has received much less attention. Even for other languages such as Japanese, existing benchmarks such as llm-jp-eval (Han et al., 2024) tend to test these higher-level abilities in Japanese, and do not test any Japanese-specific abilities. This is concerning, because Japanese contains several rare grammatical structures that require special attention, especially considering that many foundation models are now trained on multilingual datasets, and one would assume some degree of cross-lingual information sharing due to the prevalence of linguistic universals (Greenberg J. H., 1963). We inspect the perplexities of pretrained LLMs in order to study nuanced grammatical corner cases in Japanese, which are frequently studied in linguistics literature, but for which it would be difficult to create a large dataset. In particular, we study the "first person psych predicate restriction" with models in the 7-10B parameter range. We find that of open source models that currently exist, only Weblab has lower perplexities on grammatical sentences compared to ungrammatical sentences containing psych predicates. We provide evidence that this is due to Weblab's uniformly bad tokenizations, and show that Llama 3 performs orders of magnitude better on this task when its test set is

Figure 1: State of the art language models frequently fail to respect nuanced aspects of Japanese grammar, such as the first person psych predicate restriction, where here GPT-4o produces a sentence which is functionally identical to the ungrammatical Example (2) in Section 2.1.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 970–976
July 27 - August 1, 2025

restricted to sentences which it tokenizes uniformly well. We further show evidence that language models will, when tasked to output sentences containing third-person psych predicate expressions, utilize unusual circumlocutions or grammar patterns. These findings suggest that more consistent tokenizers for Japanese could lead to LLMs which are able to more closely obey relatively obscure Japanese grammar rules and output sentences utilizing more natural grammatical constructions.

2 Linguistic Preliminaries

2.1 First Person Psych Predicate Restriction

Japanese is notable for having what is referred to as a "first person psych predicate restriction," meaning that certain predicates which describe the internal states of people (such as "happy," "sad," "cold," "dizzy," etc.) can only be used to describe the first person in general when used directly. When describing others, evidential expressions (such as "seems," "appears," etc) must be used (Hasegawa and Hirose, 2005; Lee, 2012). Hasegawa and Hirose (2005) give the following examples:

(1) Watashi wa samu-i.
I TOP feel.cold-NPST
"I feel cold."

(2) #Haha wa samu-i.
Mother TOP feel.cold-NPST
"My mother feels cold." [Intended]

(3) Haha wa samu-gat-te i-ru.
Mother TOP feel.cold-EVID-CONJ be-NPST
Lit. "My mother is showing signs of feeling cold."

(4) Haha wa samu-soo da.
Mother TOP feel.cold-EVID COP.NPST
"My mother appears to feel cold."

Native speakers of Japanese generally obey the first-person psych predicate restriction in all cases, even if they are not consciously aware of its existence. L2 speakers of Japanese, however, are frequently unaware of it, which can lead them to produce utterances that are extremely confusing to native speakers. Similar grammatical phenomena and L2 misuse have been observed in Korean, which features a comparable first-person psych predicate restriction (Ahn and Mariano, 2024).

As Hasegawa and Hirose (2005) further note, when the predicate is polysemous, the semantic role of the subject shifts to conform to this restriction (so that Example (2) must be interpreted as "I, the mother, am cold" or possibly "I am cold because of my mother"). There are also exceptions for the nonreportive style (such as in a novel, where the narrator can know the internal states of all characters) where third-person subjects are permitted for psych predicate expressions (Kuroda, 1973). From the perspective of LLM training, checking whether or not a model obeys the first person psych predicate restriction is interesting, as we must test whether the LLM has learned to *not* do something which was *not* in the training dataset, rather than to reproduce co-occurrences and patterns which *did* exist in the dataset. Notably, even state of the art models such as GPT-4o (OpenAI, 2023) frequently create ungrammatical or strange sentences which fail to respect this grammatical rule, as shown in Figure 1.

3 Experiments

We wish to see whether language models have the capability to understand the grammar rules introduced in Sections 2.1. For this case, it is simple to create minimal pairs of sentences similar to Examples (1), (3), and (4), which can be compared against ungrammatical constructions similar to Example (2). We evaluate the Huggingface (Wolf et al., 2019) implementations of the following LLMs in the 7B to 10B parameter range: Mistral 0.1-7B (Jiang et al., 2023), Llama 2-7B (Touvron et al., 2023), Llama 3-8B (Llama Team, 2024), Weblab-10B[1], Swallow-7B (Fujii et al., 2024), and Swallow-MS-7b-v0.1[2]. Mistral, Llama 2, and Llama 3 are multilingual LLMs, whereas Weblab, Swallow, and Swallow-MS are specifically tuned for Japanese applications. In Section 3.1 we choose to use perplexity on base models as our measure because it can be easily applied on the sentence level regardless of how the sentence is tokenized, and also correlates directly with token generation probabilities (e.g., sentence x has lower perplexity than y generally implies that sentence x is more likely to be generated than y). We also report the median over perplexity scores rather than the mean as the scales and numerical values of token probabilities can differ greatly between models due to variations in training data, vocabulary sizes, and tokenizer

[1] https://huggingface.co/matsuo-lab/weblab-10b
[2] https://huggingface.co/tokyotech-llm/Swallow-MS-7b-v0.1

	Mistral	Llama 2	Llama 3	Weblab	Swallow	Swallow-MS
(#) 3rd, psych, direct	2.0e+04	3.3e+04	6.9e+03	2.0e+06	1.2e+03	1.9e+03
(a) 1st, psych, direct	3.6e+04	1.2e+05	9.1e+04	6.1e+05	1.9e+03	3.2e+03
(b) 3rd, non-psych, direct	1.8e+03	5.9e+03	4.5e+03	7.3e+05	1.2e+03	2.9e+03
(c) 3rd, psych, evidential	2.0e+04	4.9e+04	3.7e+04	1.3e+06	4.1e+03	3.3e+03

Table 1: Median perplexity over language models, for sentences corresponding to those introduced in Section 2.1. Weblab is the only model which has lower perplexities for all grammatical constructions (labeled a, b, c) relative to the ungrammatical direct third person psych predicate (labeled #), which we believe is due to its uniformly bad tokenization. Green, yellow, and red indicate perplexity for grammatical constructions that are respectively lower than, equal to, and higher than that of the grammatical constructions.

designs (including differing fertility scores, as discussed below). These differences tend to affect the mean much more significantly due to its sensitivity to outliers, making the median a more robust statistic for comparing central tendencies across diverse models in this context. In Section 3.2 we further evaluate the instruction-following variants of these models when presented with machine translation tasks.

3.1 Perplexity Experiments

For the psych predicate restriction, an ideal language model would have the following properties:

(a) The probability of an adjective or verb which is not a psych predicate given that the sentence is in the third person is higher than the probability of an adjective or verb which is a psych predicate given that the sentence is in the third person.

(b) The probability of an adjective or verb which is a psych predicate given that the sentence is in the first person is higher than the probability of the same adjective or verb given that the sentence is in the third person.

(c) The probability that a psych predicate is used with an evidential given that the sentence is in the third person is higher than the probability of the same psych predicate being used directly given that the sentence is in the third person.

We show perplexities corresponding to the grammatical constructions in (a), (b), and (c) above, as well as the ungrammatical third person direct psych predicate expression, in Table 1 with templates for our test sentences in Appendix B. Despite having much higher perplexities in general, only Weblab has lower perplexities for the grammatical constructions than the ungrammatical construction. We found that our results were influenced heavily by tokenization, particularly whether or not byte fallback (which reduces the probabilities of tokens by several orders of magnitude) was needed, and whether or not the tokenization corresponded to the underlying linguistic structure. In particular, we noticed that for Llama 3, while certain adjectives ("itai," "samui," "tsurai," "kokorobosoi," "atsui") caused well-behaved tokenizations, those ending in "shii" such as ("kanashii," "sabishii," "ureshii," "hazukashii," "kurushii") caused byte fallback when used in conjunction with evidential expressions, thus giving these tokens extremely low probability, even when they are grammatically necessary. When restricting our test sentences to only those which Llama 3 tokenizes well, we found that its median perplexity for evidentials used with psych predicates in the third person (sentence type (c)) drops from the reported 3.7e+04 to 1.3e+03 (~28x difference), whereas the perplexity of the ungrammatical construction drops from the reported 6.9e+03 to only 3.9e+03 (~1.8x difference). Consequently, the Llama 3 base model is very likely to output the ungrammatical direct form of a psych predicate adjective with a third person topic, if and only if that adjective ends in "shii" as opposed to just "i." Note that for sentence type (c) compared to the ungrammatical construction only the final tokens in the sentence differ, so lower perplexity is exactly equivalent to higher generation chance.

Ironically, Weblab may have performed best on these tasks because it was trained using a tokenizer for English models as-is, and as such does not contain tokens for even extremely common Japanese characters[3]. As a result, it tokenizes ev-

[3]Weblab's tokenizer uses byte fallback for words such as

	Mistral	Llama 2	Llama 3	Weblab	Swallow	Swallow-MS
Fertility	1.51	1.58	0.85	1.23	1.0	1.0
Byte Fallback	0.43	0.49	0.08	0.66	0.19	0.19

Table 2: Japanese fertility scores and byte fallback rates across studied models over sentences produced by the templates given in Appendix B. Due to its use of an unmodified English tokenizer, the majority of tokens produced by Weblab are out-of-vocabulary.

	Weblab				Llama 3			
	Cold	Embarrassed	Lonely	Pain	Cold	Embarrassed	Lonely	Pain
✔evidential	47	90	0	0	0	32	0	0
✔grammatical	53	10	0	6	0	0	0	0
✖no evidential	0	0	100	94	69	39	100	100
✖mistranslation	0	0	0	0	0	29	0	0
✖wrong syntax	0	0	0	0	31	0	0	0

Table 3: Weblab and Llama 3 outputs when asked to translate the English sentence "My mother is {psych predicate}" into Japanese. While Llama 3 struggled to output evidential expressions at all, Weblab was able to consistently output evidential expressions with a third person subject feeling "cold" or "embarrassed." Here "grammatical" indicates alternative phrasings that are grammatical translations of the sentence, but do not require the use of evidential expressions.

ery sentence uniformly poorly, and does not have grammar-specific tokenization issues. To quantitatively support claims about tokenizer strength, we report fertility scores (Rust et al., 2021) and byte fallback rates in Table 2. Weblab, which used an unmodified English language tokenizer, must output significantly more tokens, and outputs out-of-vocabulary tokens much more frequenly than LLMs such as Llama 3, which use tokenizers more adapted to Japanese. These findings suggest that the same issues which seem to appear in very large state of the art models, such as GPT-4o given in Figure 1, could possibly be alleviated simply by using more consistent tokenizers or, alternatively, using uniformly bad tokenizers (such as only bytes).

3.2 Machine Translation Experiments

We further conducted machine translation experiments with the instruction-following variants of Weblab and Llama 3, on the sentence "My mother is {psych predicate}" with the four psych predicates: cold, embarrassed, lonely, and pain. In these experiments, we collect 100 samples using a temperature of 0.4 to encourage a distribution of outputs which reflect the most likely outputs of each model. The authors labeled each output of the model, categorizing them into one of {evidential, grammatical (non-evidential), no evidential (ungrammatical), mistranslation, wrong syntax}, and summarized these results in Table 3.

As the perplexity experiments in Section 3.1 suggested, while Weblab was able to output evidential expressions, or otherwise grammatical expressions, fairly often, Llama 3 very rarely used evidential expressions, and failed to output grammatical expressions in most cases. Furthermore, unlike Weblab, Llama 3 also occasionally produced sentences which were semantically incorrect given the English sentence (mistranslations) or syntactically incorrect (invalid Japanese), despite Llama 3 generally outperforming Weblab by a large margin in Japanese tasks. Notably, the instruction-following Llama 3 output ungrammatical expressions even in the case in which the base model had lower perplexity for grammatical expressions relative to ungrammatical expressions (the psych predicate adjectives which do not end in "shii" such as "itai" (feels pain) and "samui" (feels cold)). In the machine translation setup, both the weights of the model and prompts used differ drastically from those in the perplexity experiments, which makes this difference unsurprising. Nevertheless, the general trend that Weblab was able to output grammatical expressions featuring evidentials when psych predicates

"taberu" (to eat) and "kau" (to buy), the characters for which are learned in the second year of elementary school.

were used in the third person, despite its weaker tokenizer and even in this vastly different scenario, points to our claim that inconsistent tokenizations are the root cause behind LLMs making this grammatical error in Japanese.

4 Conclusion

We constructed a set of minimal pair test sentences to measure the Japanese grammar abilities of several open source models, particularly in their ability to recognize the "first person psych predicate restriction." For these minimal pairs, we showed that inconsistent tokenizations cause language models to produce perplexities that do not match the grammatical rules. Only when given uniformly good, or uniformly bad, tokenizations of Japanese were these models able to produce lower perplexity for grammatical psych predicate constructions relative to ungrammatical psych predicate constructions. Our findings should inform future constructions of tokenizers and pretraining datasets, which ultimately should lead to future language models being able to more closely follow nuanced grammar rules in Japanese and other languages.

5 Limitations

While we have done our best to come up with explanations for the phenomena which we have observed, the language models studied have many confounding factors and unknowns which could be altering token probabilities in ways that we could not know. For instance, in general we cannot not know the exact effects of the amount of Japanese pretraining data, the ratio of Japanese pretraining data, and the effect of tokenizations on the produced token probabilities. While the existence of byte fallback played a crucial role in our analysis, it also serves as a confounding factor, as it causes single Japanese characters to be split into multiple tokens, and makes the probabilities for those tokens extremely low.

References

Hyunah Ahn and Gemmalyn Mariano. 2024. The Morphological Distribution of English Two-place Psych-predicates in Children's Language Input. *Yeongeo Yeongmunhak Yeongu (Journal of the English Language and Literature)*, 50.4:219–237.

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual Pre-Training for Cross-Lingual LLM Adaptation: Enhancing Japanese Language Capabilities. In *First Conference on Language Modeling*.

Greenberg J. H. 1963. Some universals of grammar with particular reference to the order of meaningful elements. In *Universals of Language*.

Namgi Han, Nobuhiro Ueda, Masatoshi Otake, Satoru Katsumata, Keisuke Kamata, Hirokazu Kiyomaru, Takashi Kodama, Saku Sugawara, Bowen Chen, Hiroshi Matuda, Yusuke Miyao, Yugo Miyawake, and Koki Ryu. 2024. llm-jp-eval: Automatic evaluation tool for Japanese large language models. In *The 30th Annual Meeting of Japanese Association for Natural Language Processing (NLP2024)*.

Yoko Hasegawa and Yukio Hirose. 2005. What the Japanese language tells us about the alleged Japanese relational self. *Australian Journal of Linguistics*, 25(2).

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Sige-Yuki Kuroda. 1973. Where Epistemology, Style, and Grammar Meet: A Case Study from Japanese. In Stephen R. Anderson and Paul Kiparsky, editors, *A Festschrift for Morris Halle*, pages 377–391. New York: Holt, Rinehart & Winston.

Chungmin Lee. 2012. Psych-predicates: How they are different. In *Proceedings of the 26th Pacific Asia Conference on Language, Information and Computation, PACLIC 2012*.

Llama Team. 2024. The Llama 3 Herd of Models. Technical report.

OpenAI. 2023. GPT-4 Technical Report. Technical report.

Phillip Rust, Jonas Pfeiffer, Ivan Vulic, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? On the monolingual performance of multilingual language models. In *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and others. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

A Interlinear Notation Key

ACC	accusative
COP	copula
CONJ	conjunctive
EVID	evidential
GEN	genitive
LAT	lative
NMLZ	nominalizer
NPST	nonpast tense
NOM	nominal
PST	past
TOP	topic marker

B Section 3.1 Psych Predicate Experiment Template

The template used to generate sentences for the psych predicate experiments is as follows:

(5) {Watashi/Haha} wa
{I/Mother} TOP
{predicate}-{i/soo}.
{predicate}-{NPST/EVID}
"{I/My mother} {feel(s)/appears to feel} {predicate}."

B.1 Python Code for Dataset Generation

The Python code in Figure 2 can be used to generate the example sentences based on the template structure.

B.2 Predicates Used in Section 3.1 Psych Predicate Experiments

yasashi	nice
kawai	cute
oso	late
haya	fast
chika	close
waka	young
omoshiro	interesting
segataka	tall
urusa	loud
tsuyo	strong
ita	feel.pain
samu	feel.cold
tsura	feel.difficult
kokoroboso	feel.helpless
atsu	feel.hot
kanashi	feel.sad
sabishi	feel.lonely
ureshi	feel.happy
hazukashi	feel.embarrassed
kurushi	feel.strenuous

```python
def generate_sentences() -> None:
    subjects = {
        "watashi": "私",  # I
        "haha": "母"     # Mother
    }

    # Psych predicates and non-psych predicates
    predicates = {
        "yasashii": "優しい",      # nice (non-psych)
        "kawaii": "可愛い",        # cute (non-psych)
        "osoi": "遅い",            # late/slow (non-psych)
        "hayai": "速い",           # fast (non-psych)
        "chikai": "近い",          # close (non-psych)
        "wakai": "若い",           # young (non-psych)
        "omoshiroi": "面白い",     # interesting (non-psych)
        "segatakai": "背が高い",  # tall (non-psych)
        "urusai": "うるさい",      # loud/annoying (non-psych)
        "tsuyoi": "強い",          # strong (non-psych)
        "itai": "痛い",            # feel.pain (psych)
        "samui": "寒い",           # feel.cold (psych)
        "tsurai": "辛い",          # feel.difficult/painful (psych)
        "kokorobosoi": "心細い",  # feel.helpless (psych)
        "atsui": "暑い",           # feel.hot (psych)
        "kanashii": "悲しい",      # feel.sad (psych)
        "sabishii": "寂しい",      # feel.lonely (psych)
        "ureshii": "嬉しい",       # feel.happy (psych)
        "hazukashii": "恥ずかしい", # feel.embarrassed (psych)
        "kurushii": "苦しい"       # feel.strenuous/painful (psych)
    }

    for subj_key, subject_jp in subjects.items():
        for pred_key, predicate_jp in predicates.items():
            # Generate direct form (-i)
            # Grammatical for 1st person (watashi) + any predicate
            # Grammatical for 3rd person (haha) + non-psych predicate
            # Ungrammatical for 3rd person (haha) + psych predicate
            print(f"{subject_jp}は{predicate_jp}")

            # Generate evidential form (-sou)
            if predicate_jp.endswith("い"):
                 # Adjective stem + そう
                sou_form = predicate_jp[:-1] + "そう"
                print(f"{subject_jp}は{sou_form}")

if __name__ == "__main__":
    generate_sentences()
```

Figure 2: Python code to generate example sentences.

Unique Hard Attention: A Tale of Two Sides

Selim Jerad **Anej Svete** **Jiaoda Li** **Ryan Cotterell**
{sjerad, anej.svete, jiaoda.li, ryan.cotterell}@ethz.ch
ETH zürich

"A wonderful fact to reflect upon, that leftmost and rightmost unique hard attention are constituted to be profoundly distinct."

Abstract

Understanding the expressive power of transformers has recently attracted attention, as it offers insights into their abilities and limitations. Many studies analyze unique hard attention transformers, where attention selects a single position that maximizes the attention scores. When multiple positions achieve the maximum score, either the rightmost or the leftmost of those is chosen. In this paper, we highlight the importance of this seeming triviality. Recently, finite-precision transformers with both leftmost- and rightmost-hard attention were shown to be equivalent to Linear Temporal Logic (**LTL**). We show that this no longer holds with only leftmost-hard attention—in that case, they correspond to a *strictly weaker* fragment of **LTL**. Furthermore, we show that models with leftmost-hard attention are equivalent to *soft* attention, suggesting they may better approximate real-world transformers than right-attention models. These findings refine the landscape of transformer expressivity and underscore the role of attention directionality.

1 Introduction

Much work has recently been done on understanding the capabilities and limitations of transformers (Vaswani et al., 2017). Collectively, the body of work on the representational capacity of transformers has provided a nuanced picture of the landscape (Pérez et al., 2021; Hahn, 2020; Chiang and Cholak, 2022; Hao et al., 2022; Merrill et al., 2022a; Merrill and Sabharwal, 2023; Chiang et al., 2023; Yang et al., 2024a; Strobl et al., 2024; Svete and Cotterell, 2024; Nowak et al., 2024; Li and Cotterell, 2025, *inter alia*).[1] Any such investigation begins by choosing an idealization of the architecture. A common modeling decision is to use **unique hard attention** (UHA), which selects a single position maximizing the attention scores (Hahn, 2020; Hao et al., 2022; Barceló et al., 2024). In one of the first exact descriptions of UHA expressivity, Yang et al. (2024a) show that UHA transformers (UHATs) with no positional encodings, strict future masking, and *either leftmost* or *rightmost* tiebreaking are equivalent to linear temporal logic, **LTL**. This connects UHATs to well-understood formalism such as the star-free languages and counter-free automata. An incautious reading of Yang et al.'s result could lead one to generalize the equivalence to all UHATs. However, we zoom in on the overlooked choice of tiebreaking and show that it markedly impacts the model's expressivity.

We show UHATs with *only leftmost* tiebreaking are strictly less expressive by relating them to a fragment of **LTL**, $\mathbf{LTL}[\diamondminus]$. We do so by adapting Yang et al.'s (2024a) proofs: We describe a variant of the **B-RASP** programming language defined therein, restricted to leftmost tiebreaking, and show it to be equivalent to $\mathbf{LTL}[\diamondminus]$. Further, leveraging the recent results by Li and Cotterell (2025) that characterize finite-precision future-masked *soft*-attention transformers as equivalent to $\mathbf{LTL}[\diamondminus]$, we establish the equivalence of leftmost UHATs to standard softmax transformers. Moreover, we give explicit $\mathbf{LTL}[\diamondminus]$ formulas and partially ordered finite-state automata (FSAs) describing leftmost UHATs.

2 Transformer Idealization

This section introduces the idealization of the transformer analyzed throughout the paper.

Finite Precision. Implemented on modern hardware, computations performed by transformers rely on a fixed number of bits. This makes finite-precision a more realistic assumption than unbounded or growing (w.r.t. input length) precision (Barceló et al., 2024; Hahn, 2020; Hao et al., 2022; Merrill et al., 2022b; Merrill and Sabharwal, 2023).

[1] App. A holds a more detailed overview of related work.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 977–996
July 27 - August 1, 2025

No positional encodings. We wish to study what we consider to be a *barebone* idealization of the transformer, because this enables us to understand the exact expressive power of UHA. Moreover, finite-precision positional encodings correspond precisely to monadic predicates in **LTL** formulas (Yang et al., 2024a), yielding a predictable and well-understood extension to our analysis.

Unique hard attention. While the original transformer uses soft attention (Vaswani et al., 2017), theoretical work largely analyzes hard attention (Merrill et al., 2022b; Hao et al., 2022; Yang et al., 2024a; Barceló et al., 2024, *inter alia*). The precise implications of this modeling decision are still unclear, but our work, combined with Li and Cotterell's (2025) results, reduces the gap between both models: Soft-attention transformers are equivalent to leftmost UHATs while rightmost UHATs are more expressive. We contextualize our findings with some more related results in Tab. 1.

Strict future masking. Future masking is standard in transformer-based language models. We focus on *strict* masking (where a position cannot attend to itself) as non-strict masking is known to reduce expressive power (Yang et al., 2024a). Moreover, residual connections still allow the model to incorporate information vertically across layers.

3 The Best of UHA, The Worst of UHA

This section provides a high-level overview, intuition, roadmap, and key implications of our results.

3.1 Separation

We begin by building an intuition as to why, with future masking, UHA with leftmost tiebreaking $\blacktriangleleft$ is strictly less expressive than with rightmost tiebreaking $\blacktriangleright$. We illustrate this in **B-RASP**, a Boolean-valued programming language (Yang et al., 2024a), as the intermediary between **LTL** and UHATs. To follow the coming examples, we only need familiarity with the following attention operation:

$$P(t) = \blacktriangleleft_{t'}[t' < t, s(t')]\ v(t') : d(t) \qquad (1)$$

$\blacktriangleleft_{t'}[t' < t, s(t')]$ denotes choosing (*attending to*) the *leftmost* ($\blacktriangleleft$) position t' for which $t' < t$ holds (Future masking) and $s(t') = 1$. If such a t' exists, $P(t)$ is assigned the value of the predicate $v(t')$, and otherwise, it is assigned a default value $d(t)$. This emulates leftmost UHA, where $t' < t$ corresponds to strict future masking, $s(t') = 1$ corresponds to maximizing the attention score, and $v(t')$ corresponds to the value vector. We define the rightmost operation $\blacktriangleright$ analogously.

We now note two facts:

(i) Every leftmost operation $\blacktriangleleft$

$$P(t) = \blacktriangleleft_{t'}[t' < t, s(t')]\ v(t') : d(t) \qquad (2)$$

can be simulated by $\blacktriangleright$ attentions in two steps. First, we gather all positions t that have preceding positions $t' < t$ such that $s(t') = 1$:

$$P_1(t) = \blacktriangleright_{t'}[t' < t, s(t')]\ s(t') : 0. \qquad (3)$$

Then, the leftmost position is just the single position t' with $s(t') = 1$ but $P_1(t') = 0$:

$$P_2(t) = \blacktriangleright_{t'}[t' < t, s(t') \wedge \neg P_1(t')]\ v(t') : d(t). \qquad (4)$$

(ii) There exist operations that $\blacktriangleright$ *can* perform and $\blacktriangleleft$ *cannot*. For instance, $\blacktriangleright$ attention can read the value immediately to the left of the current position, i.e., $v(t-1)$, as follows:

$$P(t) = \blacktriangleright_{t'}[t' < t, 1]\ v(t') : 0, \qquad (5)$$

but $\blacktriangleleft$ attention cannot, as we would need $t' = t-1$ to be the only position with $t' < t$ and $s(t') = 1$ for all $t \in [T]$, which is impossible.

This establishes a separation between **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$, which is limited to leftmost tiebreaking $\blacktriangleleft$ and future masking F, and the full **B-RASP**, leading to the following:

Theorem 3.1 (Informal)**.** *Finite-precision future-masked $\blacktriangleleft$ UHATs are* weaker *than $\blacktriangleright$ UHATs.*

3.2 Characterizations

In the remainder of the paper, we show that **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$ is equivalent to the fragment of **LTL** with only the $\diamondminus$ operator, denoted by **LTL**$[\diamondminus]$, which in turn is equivalent to partially ordered FSAs (POFA). This provides an exact characterization of leftmost UHATs:

Theorem 3.2 (Informal)**.** *Finite-precision future-masked $\blacktriangleleft$ UHATs are equivalent to* **LTL**$[\diamondminus]$.

In §6, we formalize the theorem with proofs intentionally made analogous to Yang et al.'s (2024a), in order to highlight the difference between $\blacktriangleleft$ and $\blacktriangleright$. Additionally, in §7, we provide alternative proofs that directly translate $\blacktriangleleft$ UHATs to **LTL**$[\diamondminus]$ formulas and POFA.

Transformer	**LTL**	FO logic	Regex	Monoid	Automata	Note
$\mathcal{T}^{\mathsf{F}}_{\blacktriangleright}, \mathcal{T}^{\mathsf{P}}_{\blacktriangleleft}$	$\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$	$\mathbf{FO}[<]$	star-free	aperiodic	counter-free	Yang et al. (2024a, Thm. 5)
future-masked *soft* attention	$\mathbf{LTL}[\diamondminus]$	$\mathbf{PFO}^2[<]$	$\mathcal{R}$-expression	$\mathcal{R}$-trivial	POFA	Li and Cotterell (2025)
$\mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$	$\mathbf{LTL}[\diamondminus]$	$\mathbf{PFO}^2[<]$	$\mathcal{R}$-expression	$\mathcal{R}$-trivial	POFA	Thms. 6.1 and 6.2
$\mathcal{T}^{\mathsf{P}}_{\blacktriangleright}$	$\mathbf{LTL}[\diamondplus]$	$\mathbf{FFO}^2[<]$	$\mathcal{L}$-expression	$\mathcal{L}$-trivial	RPOFA	Thm. E.1

Table 1: Known equivalences of finite-precision transformers with no positional encodings to different formalism. $\mathcal{T}^{\mathsf{F}}_{\blacktriangleright}$ future-masked rightmost UHATs. $\mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$, $\mathcal{T}^{\mathsf{P}}_{\blacktriangleleft}$, and $\mathcal{T}^{\mathsf{P}}_{\blacktriangleright}$ are defined analogously for past masking and leftmost UHA.

3.3 Implications

Combining with Li and Cotterell's (2025) results that show that soft and average hard attention transformers[2] are equivalent to $\mathbf{LTL}[\diamondminus]$ as well, we discover the peculiar fact that with fixed precision, soft attention, average hard attention, and leftmost UHA are all equally expressive.

This insight could shed light on certain observed phenomena in soft-attention transformers. For instance, Liu et al. (2023) find that transformers struggle with the flip-flop language, where the symbol following a "read" instruction must match the symbol following the most recent "write" instruction. Our results suggest that this difficulty arises because leftmost UHATs and thereby soft-attention transformers lack the ability to locate the most recent—rightmost—write instruction.

Furthermore, the fact that rightmost UHA is strictly more expressive than other variants of transformers may partly explain the empirical success of positional encodings that bias attention toward recent tokens, such as ALiBi (Press et al., 2022), as they help approximate rightmost tiebreaking.

4 Linear Temporal Logic

An **alphabet** Σ is a finite, non-empty set of **symbols**. The **Kleene closure** of Σ is $\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$, the set of all strings, where $\Sigma^0 \stackrel{\text{def}}{=} \{\varepsilon\}$ contains only the empty string. A **language** $\mathbb{L}$ is a subset of Σ^*. We treat a **language recognizer** as a function $\mathtt{R}\colon \Sigma^* \to \{0, 1\}$ whose language is $\mathbb{L}(\mathtt{R}) \stackrel{\text{def}}{=} \{\boldsymbol{w} \in \Sigma^* \mid \mathtt{R}(\boldsymbol{w}) = 1\}$. Two recognizers $\mathtt{R}_1$ and $\mathtt{R}_2$ are **equivalent** if and only if $\mathbb{L}(\mathtt{R}_1) = \mathbb{L}(\mathtt{R}_2)$.

Linear temporal logic $\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$ is an extension of Boolean logic that considers events over time and can express time-dependent properties (Pnueli, 1977). We define **LTL** over (finite) strings. Formulas in $\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$ are composed of atomic formulas π_a for $a \in \Sigma$, Boolean connectives $\wedge, \neg$,[3] and four **temporal operators** $\diamondminus$ (past), $\diamondplus$ (future), $\mathbf{S}$ (since) and $\mathbf{U}$ (until). We denote by $\mathbf{LTL}[\diamondminus]$ the fragment with only the $\diamondminus$ operator and Boolean connectives and by $\mathbf{LTL}[\diamondplus]$ the fragment with only the $\diamondplus$ operator and Boolean connectives.

Given a string $\boldsymbol{w} = w_1 \cdots w_T$, **LTL** formulas are interpreted at some position $t \in [T]$. We write $\boldsymbol{w}, t \models \psi$ to denote that ψ is TRUE on $\boldsymbol{w}$ at position t. The semantics for $\mathbf{LTL}[\diamondminus]$ are:[4]

- $\boldsymbol{w}, t \models \pi_a \iff w_t = a$;
- $\boldsymbol{w}, t \models \psi_1 \wedge \psi_2 \iff \boldsymbol{w}, t \models \psi_1 \wedge \boldsymbol{w}, t \models \psi_2$;
- $\boldsymbol{w}, t \models \neg\psi \iff \boldsymbol{w}, t \not\models \psi$;
- $\boldsymbol{w}, t \models \diamondminus\psi \iff \exists t' < t : \boldsymbol{w}, t' \models \psi$.

To define string acceptance, we denote by $T + 1$ a position outside of the string and define

$$\boldsymbol{w} \models \psi \Leftrightarrow \boldsymbol{w}, T + 1 \models \psi. \tag{6}$$

5 B-RASP and B-RASP$^{\mathsf{F}}_{\blacktriangleleft}$

We now introduce **B-RASP** in more detail. The input to a **B-RASP** program is a string $\boldsymbol{w} \in \Sigma^*$ with $|\boldsymbol{w}| = T$. On such an input, a **B-RASP program** computes a sequence of Boolean vectors of size T, with entries indexed by $t \in [T]$ in parentheses, denoted as $P(t)$. Each $w \in \Sigma$ gives rise to an atomic Boolean vector Q_w, defined as follows. For each $t \in [T]$:

$$Q_w(t) = 1 \iff \boldsymbol{w}_t = w \tag{7}$$

To streamline notation, we denote the first $|\Sigma|$ vectors of the program Q_w by $P_1, \cdots, P_{|\Sigma|}$. The $(i+1)^{\text{th}}$ vector $P_{i+1} \stackrel{\text{def}}{=} P'$ is computed inductively using one of the following operations:

(1) **Position-wise operation**: $P'(t)$ is a Boolean combination of zero or more of $\{P_{i'}(t)\}_{i'=1}^{i}$.

[2] Average hard attention divides attention mass equally among positions maximizing the attention score.

[3] $\vee$ can be defined using $\wedge$ and $\neg$.

[4] We refer to App. B for more details on $\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$ and its equivalent formalisms.

(2) **Attention operation**: $P'(t)$ can be one of:

$$P'(t) = \blacktriangleleft_{t'}[m(t,t'), s(t,t')]\ v(t,t') : d(t) \quad \text{(8a)}$$

$$P'(t) = \blacktriangleright_{t'}[m(t,t'), s(t,t')]\ v(t,t') : d(t) \quad \text{(8b)}$$

where:

- The **mask predicate** m is defined as either $m(t,t') \stackrel{\text{def}}{=} \mathbb{1}\{t' < t\}$ for **strict future masking** (F), or $m(t,t') \stackrel{\text{def}}{=} \mathbb{1}\{t' > t\}$ for **strict past masking** (P). Notice that the inequalities are *strict*, meaning the current position is excluded from attention. This detail has been shown to *increase* expressivity compared to non-strict masking (Yang et al., 2024a; Li and Cotterell, 2025).[5]
- The **score predicate** $s(t,t')$ is a Boolean combination of $\{P_{i'}(t)\}_{i'=1}^{i} \cup \{P_{i'}(t')\}_{i'=1}^{i}$.
- the **value predicate** $v(t,t')$ is defined analogously, and
- the **default value predicate** $d(t)$ is a Boolean combination of values in $\{P_1(t), \ldots, P_i(t)\}$.

We use $\blacktriangleleft$ to denote **leftmost** tiebreaking and $\blacktriangleright$ to denote **rightmost** tiebreaking. For $t \in [T]$, define the set of valid positions as:

$$\mathcal{N}(t) \stackrel{\text{def}}{=} \{t' \in [T] \mid m(t,t') = 1 \text{ and } s(t,t') = 1\}. \quad (9)$$

The unique position to attend to is then selected as:

$$t^* \stackrel{\text{def}}{=} \begin{cases} \min \mathcal{N}(t) & \textbf{if } \blacktriangleleft \\ \max \mathcal{N}(t) & \textbf{if } \blacktriangleright \end{cases}. \quad (10)$$

Finally, the semantics of the attention operation are given by:

$$P'(t) \stackrel{\text{def}}{=} \begin{cases} v(t,t^*) & \textbf{if } |\mathcal{N}(t)| > 0 \\ d(t) & \textbf{otherwise} \end{cases}. \quad (11)$$

Note that, by Lem. 12 and Prop. 11 of Yang et al. (2024a), we can rewrite every $\mathcal{P}$ to an equivalent program where every attention operation only uses unary scores and unary values, i.e., $s(t,t')$ and $v(t,t')$ depend only on t'.

B-RASP$^{\mathsf{F}}_{\blacktriangleleft}$ is the restricted version of **B-RASP** with only leftmost tiebreaking and future masking.

To define $\mathcal{P}$'s language, we designate a final output vector Y and T as the output position such that $Y(T) = 1$ signals acceptance.

B-RASP is equivalent to finite-precision future-masked rightmost UHATs, $\mathcal{T}^{\mathsf{F}}_{\blacktriangleright}$ (Yang et al., 2024a, Thms. 3 and 4). A similar claim, proved in App. C, holds for **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$ and leftmost UHATs, $\mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$.

[5] Such UHATs can still access information at the current position via the residual connection; similarly, **B-RASP** programs can do so using the default predicate d.

Theorem 5.1. *For any UHAT in $\mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$, there exists an equivalent **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$ program. For any **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$ program, there exists an equivalent UHAT in $\mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$.*

6 B-RASP$^{\mathsf{F}}_{\blacktriangleleft}$ Is Equivalent to LTL$[\diamondsuit]$

We now formalize the equivalence of **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$ and **LTL**$[\diamondsuit]$. It rests on the following two theorems. The proofs are provided in App. C. They are adapted from Yang et al. (2024a), with the differing parts highlighted in red.

Theorem 6.1. *For any formula ψ of* **LTL**$[\diamondsuit]$*, there is a **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$ program with a Boolean vector P_ψ such that, for any input $\boldsymbol{w}$ of length T and all $t \in [T]$, we have $\boldsymbol{w}, t \models \psi \iff P_\psi(t) = 1$.*

Theorem 6.2. *For any Boolean vector P of a **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$ program $\mathcal{P}$, there is a formula ψ_P of* **LTL**$[\diamondsuit]$ *such that for any input $\boldsymbol{w}$ of length T and all $t \in [T]$, we have $P(t) = 1 \iff \boldsymbol{w}, t \models \psi_P$.*

Yang et al. (2024a, Thm. 15) establish an alternative proof of the equivalence between **B-RASP** and full **LTL**$[\diamondsuit, \diamondsuit, \mathbf{S}, \mathbf{U}]$ via counter-free automata, which recognize the class of star-free languages. Analogously, we demonstrate that **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$ corresponds to POFAs, a subclass of counter-free automata that characterize **LTL**$[\diamondsuit]$. A translation from POFAs to **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$ is provided in App. C.1.

7 Direct Descriptions of $\mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$

We now describe an alternative description of $\mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$ that *directly* translates it to **LTL**$[\diamondsuit]$ and POFAs.

7.1 Describing $\mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$ with LTL$[\diamondsuit]$

In a transformer, the contextual representation at layer ℓ, $\boldsymbol{x}_t^{(\ell)}$, determines a function that computes the next representation, $\boldsymbol{x}_t^{(\ell+1)}$, given the unmasked symbols using the attention mechanism. In UHATs, this function is particularly simple: It computes $\boldsymbol{x}_t^{(\ell+1)}$ by *selecting* the symbol with the highest attention score (as per the tiebreaking mechanism), $\boldsymbol{x}_{t^*}^{(\ell)}$, and combines it with $\boldsymbol{x}_t^{(\ell)}$ via the residual connection: $\boldsymbol{x}_t^{(\ell+1)} = \boldsymbol{x}_t^{(\ell)} + \boldsymbol{x}_{t^*}^{(\ell)}$; see Fig. 1.

This invites the interpretation of transformer layers collecting progressively richer representations of individual symbols by selecting a new representation to append at each layer. We translate this idea into a set of **LTL**$[\diamondsuit]$ formulas of the form $\phi^{(\ell)}_{\boldsymbol{x}_t^{(\ell)} \leftarrow \boldsymbol{x}_{t^*}^{(\ell)}}$ that keep track of the fact that the representation $\boldsymbol{x}_t^{(\ell)}$ was updated with the representation

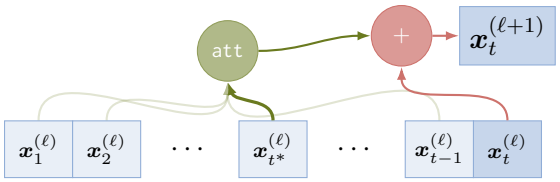


Figure 1: Unique hard attention. $\boldsymbol{x}_{t*}^{(\ell)}$ is combined with $\boldsymbol{x}_t^{(\ell)}$ to compute $\boldsymbol{x}_t^{(\ell+1)} = \boldsymbol{x}_t^{(\ell)} + \boldsymbol{x}_{t*}^{(\ell)}$.

$\boldsymbol{x}_{t*}^{(\ell)}$ at layer ℓ. The full formula is presented in the proof of the following theorem in App. D.2.

Theorem 7.1. *Let $T \in \mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ be a transformer. Then, there exists an equivalent formula $\psi \in$ **LTL**$[\diamondminus]$.*

7.2 Describing $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ with POFAs

To provide an automata-theoretic take on the result, we directly express $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ transformers with POFAs in the proof of the following theorem in App. D.3.

Theorem 7.2. *Let $T \in \mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ be a transformer. Then, there exists an equivalent POFA.*

8 Discussion and Conclusion

We establish the equivalence of future-masked finite-precision leftmost UHATs with no positional encodings, $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$, to a fragment of linear temporal logic, **LTL**$[\diamondminus]$. Together with Yang et al.'s and Li and Cotterell's results, this largely completes the picture of finite-precision transformer expressivity.

Equivalence to soft-attention transformers. §6 not only provides a characterization of $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ in terms of **LTL**$[\diamondminus]$, but also establishes its equivalence to future-masked, finite-precision *soft*-attention transformers, in conjunction with the results of Li and Cotterell's (2025) (summarized in Tab. 1). This equivalence yields a compelling interpretation of $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ as a principled abstraction of soft-attention transformers—one that is more appropriate than $\mathcal{T}_{\blacktriangleright}^{\mathsf{F}}$ due to the expressivity gap between soft attention and rightmost UHA. Consequently, this motivates further investigation of $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ as a simplified yet faithful analog of more complex soft-attention architectures.

Dot-depth hierarchy. The **dot-depth hierarchy** (Cohen and Brzozowski, 1971) classifies star-free languages based on the minimal alternation depth of *concatenation* and *Boolean operations* in the regular expressions defining them. The hierarchy is infinite and reflects increasing expressive power. Brzozowski and Ellen (1980) show that the class of languages definable in **LTL**$[\diamondminus]$ forms a strict subclass of dot-depth 2, while being incomparable to dot-depth 1. In a related line of work, Bhattamishra et al. (2020) empirically find that transformers struggle to generalize to star-free languages with dot-depth greater than 1.

Until Hierarchy. An alternative (infinite) hierarchy spanning the star-free languages is that of **until hierarchy** (Etessami and Wilke, 1996; Thérien and Wilke, 2001), which stratifies the family according to the required number of **U** (or equivalently, **S**) operations in an **LTL** formula required to define a language. Our and Li and Cotterell's (2025) results naturally place leftmost UHA and soft-attention transformers within the 0th layer of this hierarchy.

Abilities and Limitations. Exact characterizations of transformers allow us to derive precise conclusions about models' abilities and limitations. Our results, in particular, mean that $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ transformers, like their soft-attention counterparts, *cannot* model simple languages such as *(i)* strictly local languages and n-gram models, which have been linked to infinite-precision UHATs (Svete and Cotterell, 2024); *(ii)* locally testable languages (which require detecting contiguous substrings); *(iii)* languages of nested parentheses of bounded depth (bounded Dyck languages) which have also been linked to infinite-precision transformers (Yao et al., 2021); and *(iv)* any non-star-free languages such as PARITY. In contrast, the equivalence to **LTL**$[\diamondminus]$ means that $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ *can* model simple languages such as those whose string membership depends on the presence of not necessarily contiguous subsequences or on the string prefix. Li and Cotterell (2025) find strong empirical evidence that this theoretical equivalence faithfully translates into the practical performance of trained transformers on formal languages.

A duality. We finally note that *past* masking and *rightmost* UHA has a natural characterization with **LTL**$[\diamondplus]$, the fragment of **LTL** with only the $\diamondplus$ operator. This duality is summarized in Tab. 1 and elaborated on in App. E.

Limitations

We limit ourselves to a purely theoretical investigation of the expressivity of a particular model of a transformer. In particular, our results hold for finite-precision transformers with leftmost hard

attention and no positional encodings. This is important to consider as the representational capacity of transformers depends on the choices made for all these components. We do not consider learnability and training, which are opening up to be an exciting area of study and promise to bring our understanding of the representational capacity of transformers closer to what we observe in practice (Hahn and Rofin, 2024). In fact, due to the discrete nature of the hard-attention mechanism, training $\mathsf{UHAT}s$ in practice is infeasible. Nevertheless, recent work shows how temperature scaling and unbounded positional encodings can be used to simulate hard attention with soft attention (Yang et al., 2024b). We leave the empirical investigation of contrasting the performance of $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}, \mathcal{T}_{\blacktriangleright}^{\mathsf{F}}, \mathcal{T}_{\blacktriangleright}^{\mathsf{P}}$ on various language classes to future work.

Ethical considerations

We used AI-based tools (ChatGPT and GitHub Copilot) for writing assistance. We used the tools in compliance with the ACL Policy on the Use of AI Writing Assistance.

Acknowledgements

We would like to thank Andy Yang and Michael Hahn for useful discussions and feedback on early versions of this work. Anej Svete and Jiaoda Li are supported by the ETH AI Center Doctoral Fellowship.

References

Pablo Barceló, Alexander Kozachinskiy, Anthony Widjaja Lin, and Vladimir Podolskii. 2024. Logical languages accepted by transformer encoders with hard attention. In *The Twelfth International Conference on Learning Representations*.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. 2020. On the Ability and Limitations of Transformers to Recognize Formal Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7096–7116, Online. Association for Computational Linguistics.

Janusz Antoni Brzozowski and Faith Ellen. 1980. Languages of R-trivial monoids. *Journal of Computer and System Sciences*, 20(1):32–49.

Ashok K. Chandra, Larry Stockmeyer, and Uzi Vishkin. 1984. Constant depth reducibility. *SIAM Journal on Computing*, 13(2):423–439.

David Chiang and Peter Cholak. 2022. Overcoming a theoretical limitation of self-attention. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7654–7664, Dublin, Ireland. Association for Computational Linguistics.

David Chiang, Peter Cholak, and Anand Pillay. 2023. Tighter bounds on the expressivity of transformer encoders. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Rina Cohen and Janusz Antoni Brzozowski Brzozowski. 1971. Dot-depth of star-free events. *Journal of Computer and System Sciences*, 5(1):1–16.

K. Etessami and T. Wilke. 1996. An Until hierarchy for temporal logic. In *Proceedings 11th Annual IEEE Symposium on Logic in Computer Science*, pages 108–117.

Dov Gabbay, Amir Pnueli, Saharon Shelah, and Jonathan Stavi. 1980. On the temporal analysis of fairness. In *Proceedings of the 7th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '80, page 163–173, New York, NY, USA. Association for Computing Machinery.

Michael Hahn. 2020. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171.

Michael Hahn and Mark Rofin. 2024. Why are sensitive functions hard for transformers? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14973–15008, Bangkok, Thailand. Association for Computational Linguistics.

Yiding Hao, Dana Angluin, and Robert Frank. 2022. Formal language recognition by hard attention transformers: Perspectives from circuit complexity. *Transactions of the Association for Computational Linguistics*, 10:800–810.

William Hesse. 2001. Division is in uniform TC0. In *Automata, Languages and Programming*, pages 104–114, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hans Kamp. 1968. *Tense Logic and the Theory of Linear Order*. Ph.D. thesis, UCLA.

Kenneth Krohn and John Rhodes. 1965. Algebraic theory of machines. I. Prime decomposition theorem for finite semigroups and machines. *Transactions of The American Mathematical Society - TRANS AMER MATH SOC*, 116.

Jiaoda Li and Ryan Cotterell. 2025. Characterizing the expressivity of transformer language models. *arXiv preprint arXiv:2505.23623*.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. 2023. Exposing attention glitches with flip-flop language modeling. In

Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Robert McNaughton and Seymour Papert. 1971. *Counter-Free Automata*. M.I.T. Press research monographs. M.I.T. Press.

William Merrill and Ashish Sabharwal. 2023. A logic for expressing log-precision transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*.

William Merrill, Ashish Sabharwal, and Noah A. Smith. 2022a. Saturated transformers are constant-depth threshold circuits. *Transactions of the Association for Computational Linguistics*, 10:843–856.

William Merrill, Ashish Sabharwal, and Noah A. Smith. 2022b. Saturated transformers are constant-depth threshold circuits. *Transactions of the Association for Computational Linguistics*, 10:843–856.

Franz Nowak, Anej Svete, Alexandra Butoi, and Ryan Cotterell. 2024. On the representational capacity of neural language models with chain-of-thought reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12510–12548, Bangkok, Thailand. Association for Computational Linguistics.

Jorge Pérez, Pablo Barceló, and Javier Marinkovic. 2021. Attention is turing-complete. *Journal of Machine Learning Research*, 22(75):1–35.

Amir Pnueli. 1977. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 46–57.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

Lena Strobl. 2023. Average-hard attention transformers are constant-depth uniform threshold circuits. *arXiv preprint arXiv:2308.03212*.

Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. 2024. What formal languages can transformers express? a survey. *Transactions of the Association for Computational Linguistics*, 12:543–561.

Anej Svete and Ryan Cotterell. 2024. Transformers can represent n-gram language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6845–6881, Mexico City, Mexico. Association for Computational Linguistics.

Denis Thérien and Thomas Wilke. 2001. Temporal Logic and Semidirect Products: An Effective Characterization of the Until Hierarchy. *SIAM Journal on Computing*, 31(3):777–798.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Andy Yang, David Chiang, and Dana Angluin. 2024a. Masked hard-attention transformers recognize exactly the star-free languages. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Andy Yang, Lena Strobl, David Chiang, and Dana Angluin. 2024b. Simulating hard attention using soft attention.

Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. 2021. Self-attention networks can process bounded hierarchical languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3770–3785, Online. Association for Computational Linguistics.

A Related Work

Existing work has established a rich landscape of results on the expressivity of transformers.

Lower and upper bounds for UHA. Hahn (2020) show that UHA transformers with unbounded precision, left attention, and no masking can not recognize PARITY (bit strings with an odd number of ones) nor DYCK-1 (language of correctly nested parentheses of one type). Hao et al. (2022) refine this result by showing that UHA transformers with unbounded precision and left attention can recognize at most languages in AC^0, the family of circuits of constant depth, polynomial size, and unbounded fan-in. Maybe surprisingly, this suggests such transformers can not recognize even simple languages outside AC^0 such as PARITY and MAJORITY (all bit strings in which more than half of bits are 1s). Other problems not in AC^0 include sorting, integer multiplication (Chandra et al., 1984), and integer division (Hesse, 2001). Barceló et al. (2024) show that UHA transformers augmented with arbitrary positional encodings are lower bounded by an extension of $\mathbf{FO}[<]$ with all possible monadic numerical predicates (which includes all regular languages in AC^0). Yang et al. (2024a) further refine the understanding of the relationship between $\mathbf{FO}[<]$ and finite-precision transformers by proving the equivalence between the two when the transformers are equipped with strict future masking.

Average-hard attention. Average-hard attention (AHA) differs from UHA in that when confronted with several positions with equal scores, they average their values to compute the next contextual representation. It can be seen as a special case of soft-attention (Li and Cotterell, 2025). Hao et al. (2022) show AHA *unbounded* precision transformers are more expressive than UHA transformers, as they can recognize languages outside AC^0, such as PARITY and DYCK-1. Merrill et al. (2022b) show that AHA transformers with floating-point activations can be simulated in TC^0 (the extension of AC^0 with majority gates, which output 1 iff at least half of the inputs are 1), while Strobl (2023) extend this result by tightening the upper bound to L-UNIFORM TC^0 (which consists of TC^0 circuits with the additional constraint that there exists a deterministic Turing Machine that runs in logarithmic space that can describe the circuits).

Transformers and logic. Previous results relating transformers to logic include Chiang et al. (2023), who show that finite-precision softmax transformers are upper-bounded by a generalization of first-order logic with counting quantifiers and modular arithmetic over input position indices. On the other hand, they show this logic to be a lower bound on the expressivity of unbounded-precision transformers. Merrill and Sabharwal (2023) contribute by characterizing a more expressive variant of transformers—they allow precision logarithmic in the input length and show an upper bound of first-order logic extended with majority-vote quantifiers.

Equivalence of right-attention transformers to $\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$. Yang et al. (2024a) show the equivalence of $\mathcal{T}^{\mathsf{F}}_{\blacktriangleright}$ and $\mathcal{T}^{\mathsf{P}}_{\blacktriangleleft}$ to $\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$. In their constructions, the operators $\mathbf{S}$ and $\mathbf{U}$ in $\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$ can only be expressed by transformers in $\mathcal{T}^{\mathsf{F}}_{\blacktriangleright}$, $\mathcal{T}^{\mathsf{P}}_{\blacktriangleleft}$ respectively (and vice-verse, $\mathcal{T}^{\mathsf{F}}_{\blacktriangleright}$ and $\mathcal{T}^{\mathsf{P}}_{\blacktriangleleft}$ require the operators $\mathbf{S}$,$\mathbf{U}$ respectively in their $\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$ formulation). Moreover, by Gabbay et al. (1980), the fragment of $\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$ consisting of only $\mathbf{U}$ and the Boolean connectives, denoted by $\mathbf{LTL}[\mathbf{U}]$ (or, in the analogous symmetric case, $\mathbf{LTL}[\mathbf{S}]$), is sufficient for equivalence $\mathbf{FO}[<]$ (which is equivalent to $\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$ by Kamp (1968)).

Equivalence of soft-attention transformers to $\mathbf{LTL}[\diamondminus]$. Li and Cotterell (2025) relates $\mathbf{LTL}[\diamondminus]$ and $\mathbf{PFO}^2[<]$ (the fragment of $\mathbf{FO}[<]$ which considers at most two distinct variables at the same time where any bound variable can only peek into the past of a free variable) to languages with $\mathcal{R}$-trivial monoids (Brzozowski and Ellen, 1980). These are described by a set of equivalent formalisms such as $\mathcal{R}$-expressions, $\mathcal{R}$-trivial monoids, and partially ordered automata. Li and Cotterell (2025) use this equivalence to show strict future-masked softmax transformers are equivalent to $\mathbf{LTL}[\diamondminus]$. The upper bound is proven through the equivalent $\mathbf{PFO}^2[<]$: They show $\mathbf{PFO}^2[<]$ can simulate a sum of a finite number of fixed-precision floating-point numbers (and thus the weighted sum in softmax), while other transformations (computations of keys and queries, dot-products, etc.) can be computed with the standard Boolean connectives. They show the lower bound of $\mathbf{LTL}[\diamondminus]$ by translating the standard Boolean

operations into feedforward networks, and the $\diamond$ operation with the future-masked attention mechanism. Together with Hao et al.'s (2022) results, this illuminates the complexity of transformer expressivity: While in the unbounded-precision regime, AHA transformers strictly subsume UHA transformers, in the finite-precision regime, the relationship depends on the direction of attention mechanism. The models are equivalent if leftmost tiebreaking is used and in the case of rightmost tiebreaking, UHA is strictly more expressive than AHA, the reverse of the unbounded-precision case.

B Background

This section provides the necessary background on the formalisms used in the paper and details some concepts introduced in the main part of the paper.

B.1 Finite-State Automata

Definition B.1. *A **semiautomaton** $\mathcal{A}$ is a 3-tuple (Σ, Q, δ) where Σ is an alphabet, Q is a finite set of **states** and $\delta\colon Q \times \Sigma \to Q$ is a **transition function**.*

*We further define an **initialized semiautomaton** as a semiautomaton with an initial state.*

Definition B.2. *A **deterministic finite automaton (DFA)** $\mathcal{A}$ is a 5-tuple $(\Sigma, Q, q_\iota, F, \delta)$ where (Σ, Q, δ) is a semiautomaton, $q_\iota \in Q$ is an initial state, and $F \subseteq Q$ is a set of final states.*

Definition B.3. *Let $\delta^*\colon Q \times \Sigma^* \to Q$ be the transitive closure of δ, defined as*

$$\delta^*(q, w) = \delta(q, w), \text{ for } w \in \Sigma \tag{12a}$$

$$\delta^*(q, w_1 \cdots w_T) = \delta(\delta^*(q, w_1 \cdots w_{T-1}), w_T) \tag{12b}$$

*with $\delta^*(q, \varepsilon) = q$ for any $q \in Q$. A **partially ordered DFA (POFA)** is a DFA $\mathcal{A} = (\Sigma, Q, q_\iota, F, \delta)$ where there is a partial order relation $\preceq$ on Q defined as $q \preceq p$ if and only if $\delta^*(q, \boldsymbol{w}) = p$ for some string $\boldsymbol{w} \in \Sigma^*$.*

Intuitively, POFAs are *acyclic* DFAs with possible self-loops, resulting in partially ordered states.

Definition B.4. *Let a DFA $\mathcal{A} = (\Sigma, Q, q_\iota, F, \delta)$ and $\mathbb{L}$ be the language it accepts. We define the **reverse automaton** $\mathcal{A}^R$ as the automaton that recognizes the reverse language $\mathbb{L}^R$ consisting of all strings in $\mathbb{L}$ but reversed.*

Definition B.5. *A DFA $\mathcal{A}$ is a **partially ordered reverse automaton** (RPOFA) if $\mathcal{A}^R$ is a POFA.*

Definition B.6. *Let $\mathcal{B}_1 = (\Sigma, Q_1, \delta_1)$ be a semiautomaton. Let $\mathcal{B}_2 = (Q_1 \times \Sigma, Q_2, \delta_2)$ be another, possibly partial,[6] semiautomaton such that for every $q_1 \in Q_1, q_2 \in Q_2$, either $\delta(q_2, \langle q_1, w\rangle)$ is defined for every $w \in \Sigma$ or undefined for every $w \in \Sigma$. The **cascade product** $\mathcal{B}_1 \circ \mathcal{B}_2$ is the semiautomaton $\mathcal{C} = (\Sigma, Q, \delta)$ such that:*

- $Q = \{(q_1, q_2) : \delta_2(q_2, \langle q_1, w\rangle) \text{ is defined}\}$
- $\delta(\langle q_1, q_2\rangle, w) = (\delta_1(q_1, w), \delta_2(q_2, \langle q_1, w\rangle))$

Definition B.7. *For $n \geq 0$, an **n-way fork** is an initialized semiautomaton $(\Sigma, \{q_0, q_1, \cdots, q_n\}, \delta)$ where $\Sigma = \Sigma_0 \cup \Sigma_1 \cdots \cup \Sigma_n$, the Σ_i's are non-empty and pairwise disjoint, $\delta(q_0, a) = q_i$ for all $a \in \Sigma_i$, and $\delta(q_i, a) = q_i$ for all $a \in \Sigma$. A **half-reset** (Fig. 2) is a 1-way fork.*

Definition B.8. *A surjection $\phi\colon Q \to Q'$ is an **automaton homomorphism** from the semiautomata $\mathcal{A} = (\Sigma, Q, \delta)$ to $\mathcal{A}' = (\Sigma', Q', \delta')$ if for every $q \in Q, w \in \Sigma$:*

$$\phi(\delta(q, w)) = \delta'(\phi(q), w) \tag{13}$$

In this case, we say $\mathcal{A}'$ is homomorphic to $\mathcal{A}$, or that $\mathcal{A}'$ is the homomorphic image of $\mathcal{A}$.

[6] A partial automaton is one in which some symbol–state combinations could lead to an undefined transition.

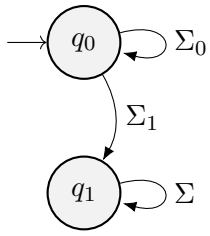


Figure 2: A 1-way fork

B.2 Syntactic Monoids

Definition B.9. *A **monoid** $\mathbb{M}$ is a set equipped with a binary operation and an identity element.*

For instance, the **free monoid** is the set Σ^* equipped with the concatenation operation and the empty string ε as identity.

Definition B.10. *A monoid $\mathbb{M}$ is **$\mathcal{R}$-trivial** if for all $w_1, w_2, w_3 \in \mathbb{M}$, $w_1 w_2 w_3 = w_1$ implies $w_1 w_2 = w_1$. A monoid $\mathbb{M}$ is **$\mathcal{L}$-trivial** if for all $w_1, w_2, w_3 \in \mathbb{M}$, $w_3 w_2 w_1 = w_1$ implies $w_2 w_1 = w_1$.*

More details about $\mathcal{R}$-trivial monoids can be found in Brzozowski and Ellen (1980).

Definition B.11. *The **syntactic congruence** $\preceq_{\mathbb{L}}$ is the equivalence relation on Σ^* given the language $\mathbb{L}$ such that for all $\boldsymbol{x}$, $\boldsymbol{y} \in \Sigma^*$, we have $\boldsymbol{x} \preceq_{\mathbb{L}} \boldsymbol{y}$ if and only if:*

$$\boldsymbol{sxz} \in \mathbb{L} \iff \boldsymbol{syz} \in \mathbb{L}\ \forall \boldsymbol{s}, \boldsymbol{z} \in \Sigma^* \tag{14}$$

Definition B.12. *The quotient monoid $\Sigma^*/\preceq_{\mathbb{L}}$ is the **syntactic monoid** of $\mathbb{L}$.*

B.3 Regular Expressions

A **regular language** can be described by **regular expressions**, which are elements of the closure of $\emptyset, \varepsilon$, and all $w \in \Sigma$ under concatenation, concatenation, and Kleene star. A regular language is **star-free** if the regular expression that describes the language does not require the Kleene star operator.

Definition B.13. *An $\mathcal{R}$-expression is a finite union of regular expressions of the form $\Sigma_0^* w_1 \Sigma_1^* \cdots w_n \Sigma_n^*$ where $w_t \in \Sigma$, $\Sigma_t \subseteq \Sigma$ and $w_t \notin \Sigma_{t-1}^*$ for $1 \leq t \leq n$. An $\mathcal{L}$-expression is defined analogously with the dual constraint of $w_t \notin \Sigma_t^*$.*

For instance, $a\Sigma^*$ is an $\mathcal{R}$-expression, while $\Sigma^* a$ is an $\mathcal{L}$-expression.

B.4 Linear Temporal Logic

We now present all possible semantics in $\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$. The semantics are defined inductively:

- $\boldsymbol{w}, t \models \pi_a \iff w_t = a$;
- $\boldsymbol{w}, t \models \psi_1 \vee \psi_2 \iff \boldsymbol{w}, t \models \psi_1 \vee \boldsymbol{w}, t \models \psi_2$;
- $\boldsymbol{w}, t \models \psi_1 \wedge \psi_2 \iff \boldsymbol{w}, t \models \psi_1 \wedge \boldsymbol{w}, t \models \psi_2$;
- $\boldsymbol{w}, t \models \neg\psi \iff \boldsymbol{w}, t \not\models \psi$;
- $\boldsymbol{w}, t \models \diamondminus\psi \iff \exists t' < t : \boldsymbol{w}, t' \models \psi$;
- $\boldsymbol{w}, t \models \diamondplus\psi \iff \exists t' > t : \boldsymbol{w}, t' \models \psi$;
- $\boldsymbol{w}, t \models \psi_1 \mathbf{S} \psi_2 \iff \exists t' < t : \boldsymbol{w}, t' \models \psi_2$ and $\boldsymbol{w}, k \models \psi_1$ for all k with $t' < k < t$;
- $\boldsymbol{w}, t \models \psi_1 \mathbf{U} \psi_2 \iff \exists t' > t : \boldsymbol{w}, t' \models \psi_2$ and $\boldsymbol{w}, k \models \psi_1$ for all k with $t < k < t'$.

$\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$ defines exactly the class of star-free languages (Kamp, 1968; McNaughton and Papert, 1971). $\mathbf{LTL}[\diamondminus]$ and $\mathbf{LTL}[\diamondplus]$, the fragments of **LTL** with only the $\diamondminus$ and $\diamondplus$ operators, respectively, are strictly less expressive than $\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$ (neither can recognize $a\Sigma^* b$, which can be recognized by $\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$), and have characterizations in terms of monoids, automata, and first-order logic (App. B.6).

B.5 First-Order Logic

First-order logic with the $<$ relation, denoted by $\mathbf{FO}[<]$, extends the usual propositional logic with predicates and quantifiers. We consider free variables $x, y, z, \cdots$ that represent positions over a string $\boldsymbol{w}$ of size T. $\mathbf{FO}[<]$ includes unary predicates π_w for $w \in \Sigma$, where $\pi_w(x) = \text{TRUE} \iff$ there is a w at position x. As with $\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$, we can inductively define formulas from π_w using the standard Boolean operators, the binary predicate $<$ between positions, and the existential quantifier $\exists$. By Kamp (1968), $\mathbf{FO}[<]$ is equivalent to $\mathbf{LTL}[\diamondminus, \diamondplus, \mathbf{S}, \mathbf{U}]$ and by McNaughton and Papert (1971), it is equivalent to the star-free languages.

$\mathbf{FO}^2[<]$ is the fragment of $\mathbf{FO}[<]$ that can only consider two distinct variables at the same time. Li and Cotterell (2025) further define $\mathbf{PFO}^2[<]$ as the past fragment of $\mathbf{FO}^2[<]$ that can only "peek into the past." Namely, any single-variable formula $\phi(x)$ can only have bounded existential quantifiers of the form "$\exists y < x$". We analogously have the future fragment of $\mathbf{FO}^2[<]$, $\mathbf{FFO}^2[<]$, where we only allow peeking into the future with quantifiers of the form "$\exists y > x$".

B.6 Equivalence Between Formalisms

Due to Brzozowski and Ellen (1980) and Li and Cotterell (2025), we have the two dual theorems:

Theorem B.1. *Let $\mathbb{L} \subseteq \Sigma^*$ be a regular language, $\mathbb{M}$ be its syntactic monoid, and $\mathcal{A}$ be the minimal DFA accepting it. The following conditions are equivalent:*

(i) $\mathbb{M}$ is $\mathcal{R}$-trivial,
(ii) $\mathbb{L}$ can be denoted by an $\mathcal{R}$-expression,
(iii) $\mathcal{A}$ is a POFA,
(iv) $\mathcal{A}$ is the homomorphic image of a cascade product of half-resets,
(v) $\mathbb{L}$ can be recognized by a formula in $\mathbf{PFO}^2[<]$,
(vi) $\mathbb{L}$ can be recognized by a formula in $\mathbf{LTL}[\diamondminus]$.

Theorem B.2. *Let $\mathbb{L} \subseteq \Sigma^*$ be a regular language, $\mathbb{M}$ be its syntactic monoid, and $\mathcal{A}$ be the minimal DFA accepting it. The following conditions are equivalent:*

(i) $\mathbb{M}$ is $\mathcal{L}$-trivial,
(ii) $\mathbb{L}$ can be denoted by an $\mathcal{L}$-expression,
(iii) $\mathcal{A}$ is a RPOFA (equivalently, $\mathcal{A}^R$ is a POFA),
(iv) $\mathcal{A}^R$ is the homomorphic image of a cascade product of half-resets,
(v) $\mathbb{L}$ can be recognized by a formula in $\mathbf{FFO}^2[<]$,
(vi) $\mathbb{L}$ can be recognized by a formula in $\mathbf{LTL}[\diamondplus]$.

C Proofs of $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ and $\mathbf{LTL}[\diamondminus]$ Equivalence

The equivalence between $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ and $\mathbf{B\text{-}RASP}_{\blacktriangleleft}^{\mathsf{F}}$ follows directly from Theorems 3 and 4 of Yang et al. (2024a).

Theorem 5.1. *For any UHAT in $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$, there exists an equivalent **B-RASP**$_{\blacktriangleleft}^{\mathsf{F}}$ program. For any **B-RASP**$_{\blacktriangleleft}^{\mathsf{F}}$ program, there exists an equivalent UHAT in $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$.*

Proof. Yang et al. (2024a, Thms. 3 and 4) and the supporting lemmata (Lemmata 21 and 24) treat $\blacktriangleleft$ and $\blacktriangleright$ separately. Translations constrained to $\blacktriangleleft$ are thus special cases of their proofs. ■

The following proofs largely follow the proofs of Lemmata 13 and 14 of Yang et al. (2024a). We highlight the part that distinguishes $\blacktriangleleft$ and $\blacktriangleright$ in red.

Theorem 6.1. *For any formula ψ of $\mathbf{LTL}[\diamondminus]$, there is a **B-RASP**$_{\blacktriangleleft}^{\mathsf{F}}$ program with a Boolean vector P_ψ such that, for any input $\boldsymbol{w}$ of length T and all $t \in [T]$, we have $\boldsymbol{w}, t \models \psi \iff P_\psi(t) = 1$.*

Proof. We proceed by induction.

Base case. The atomic formulas π_a can be represented by initial Boolean vectors Q_a for $a \in \Sigma$.

Inductive step. Assume ψ_1 and ψ_2 can be converted to **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$ vectors P_{ψ_1} and P_{ψ_2}, respectively. We distinguish three cases of building a new formula from ψ_1 and ψ_2:
• Case *(1)*: $\psi = \neg\psi_1$. Add a position-wise operation:

$$P_\psi(t) = \neg P_{\psi_1}(t). \tag{15}$$

• Case *(2)*: $\psi = \psi_1 \wedge \psi_2$. Add a position-wise operation:

$$P_\psi(t) = P_{\psi_1}(t) \wedge P_{\psi_2}(t). \tag{16}$$

• Case *(3)*: $\psi = \diamondminus\psi_1$. Add an attention operation with future masking and $\blacktriangleleft$ tiebreaking:

$$P_\psi(t) = \blacktriangleleft_{t'}[t' < t, P_{\psi_1}(t')]\ P_{\psi_1}(t') : 0. \tag{17}$$

■

Theorem 6.2. *For any Boolean vector P of a* ***B-RASP***$^{\mathsf{F}}_{\blacktriangleleft}$ *program $\mathcal{P}$, there is a formula ψ_P of* **LTL**$[\diamondminus]$ *such that for any input $\boldsymbol{w}$ of length T and all $t \in [T]$, we have $P(t) = 1 \iff \boldsymbol{w}, t \models \psi_P$.*

Proof. We proceed by induction.

Base case. Each atomic vector $Q_a(t)$ can be translated to the atomic formula π_a.

Induction step. Assume vectors $P_1, \ldots, P_{i-1}$ can be translated to $\psi_{P_1}, \ldots, \psi_{P_{i-1}}$. We distinguish different options of building a new vector out of $P_1, \ldots, P_{i-1}$:
• Case *(1)*: $P_i(t)$ is a position-wise operation:

$$P_i(t) = f(P_1(t), \ldots, P_{i-1}(t)), \tag{18}$$

where f is a Boolean function. We can translate $P_i(t)$ into $\psi_i = f(\psi_1, \ldots, \psi_{i-1})$.
• Case *(2)*: $P_i(t)$ is an attention operation that uses leftmost tiebreaking and future masking, that is,

$$P_i(t) = \blacktriangleleft_{t'}[t' < t, s(t')]\ v(t') : d(t). \tag{19}$$

By the inductive hypothesis, there are **LTL**$[\diamondminus]$ formulas ψ_S, ψ_V, and ψ_D corresponding to s, v, and d. We can thus write $P_i(i)$ as:

$$\psi_i = (\diamondminus(\psi_S \wedge \neg\diamondminus\psi_S \wedge \psi_V)) \vee (\neg(\diamondminus\psi_S) \wedge \psi_D). \tag{20}$$

where $\psi_S \wedge \neg\diamondminus\psi_S$ identifies the leftmost position that satisfies ψ_S in the string.
• Case *(3)*: We now treat the attention operation with rightmost tiebreaking and future masking, i.e.,

$$P_i(t) = \blacktriangleright_{t'}[t' < t, s(t')]\ v(t') : d(t). \tag{21}$$

In this case, we need to find the rightmost position satisfying ψ_S not in the *entire string* $\boldsymbol{w}$, but before the *current position*, which can only be realized using the **S** operator:

$$\psi_i = (\neg\psi_S \mathbf{S}(\psi_S \wedge \psi_V)) \vee (\neg(\diamondminus\psi_S) \wedge \psi_D). \tag{22}$$

■

C.1 POFAs as $\mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$

The Krohn–Rhodes theorem (Krohn and Rhodes, 1965) states that any deterministic automaton is the homomorphic image of a cascade of simple automata whose transitions induce resets or permutations of the states. Thm. B.1 by Brzozowski and Ellen (1980) provides an analogous decomposition for POFAs, namely that they are homomorphic to a cascade of *half-resets*. In a very similar manner to Yang et al. (2024a)'s Thm. 15, we can express this decomposition in a **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$ program, which can further be expressed by a transformer in $\mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$.

We first formalize how **B-RASP** can compute sequence-to-sequence functions $\Sigma^* \to \Gamma^*$ if for instance, we want a **B-RASP** program to simulate an automaton (i.e., output the same states traversed by the automaton when ran on the same input). We designate a set of output vectors Y_γ, for all $\gamma \in \Gamma$, and the sequence-to-sequence function then maps t to γ if $Y_\gamma(t)$ is true.

We then say a **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$ program $\mathcal{P}$ simulates an initialized semiautomaton $\mathcal{A} = (\Sigma, Q, \delta)$ with initial state $q_{\boldsymbol{s}}$ iff for every input string $\boldsymbol{w}$, the output vectors of $\mathcal{P}$ encode the sequence of states traversed by $\mathcal{A}$ when ran on $\boldsymbol{w}$ with initial state $q_{\boldsymbol{s}}$.

Lemma C.1. *Let $\mathcal{H} = (\Sigma_0 \cup \Sigma_1, \{q_0, q_1\}, q_0, \delta)$ be a half-reset. Then there exists a **B-RASP** program $\mathcal{P}_{\mathcal{H}}$ that simulates $\mathcal{H}$ started in start state q_0.*

Proof. We first define two **B-RASP** programs $\mathcal{B}_0$, $\mathcal{B}_1$, which return 1 if and only if the half-reset is in state q_0 or q_1, respectively. Namely:

$$\begin{aligned} \mathcal{B}_0(t) &= \blacktriangleleft_{t'}[t' < t, \bigvee_{w \in \Sigma_1} Q_w(t')]\ 0 : 1 \\ \mathcal{B}_1(t) &= 1 - \mathcal{B}_0(t) \end{aligned} \tag{23}$$

In half-resets, the presence or absence at any instance of symbols in Σ_1 determines whether we are in q_0 or q_1. It suffices to check if at the current index t, there has been a past $w \in \Sigma_1$, which can be done by computing $\bigvee_{w \in \Sigma_1} Q_w(t')$ for all indices t' satisfying future masking. ■

Lemma C.2. *Let $\mathcal{A} = (\Sigma, Q_1, \delta_1)$ be a semiautomaton that can be simulated from state $\boldsymbol{s}_1$ by a **B-RASP** program $\mathcal{P}_{\mathcal{A}}$. Let $\mathcal{H} = (Q_1 \times \Sigma, Q_2, \delta_2)$ be a half-reset and let $\mathcal{C} = \mathcal{A} \circ \mathcal{H}$. Then, there exists a program $\mathcal{P}_{\mathcal{C}}$ that simulates $\mathcal{C}$ started in state $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ for an arbitrary $\boldsymbol{s}_2 \in Q_2$.*

Proof. We now use for all $q \in Q_1$, the predicates $\mathcal{B}_{\mathcal{A},q}$ that denote whether $\mathcal{A}$ is in some state $q \in Q_1$ at time t when started at $\boldsymbol{s}_1$. By the assumption that $\mathcal{A}$ can be simulated by a **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$ program, we have access to such predicates.

We now define predicates for $q \in Q_1$, $w \in \Sigma$:

$$Q'_{(q,w)}(t) = \mathcal{B}_{\mathcal{A},q}(t) \wedge Q_w(t) \tag{24}$$

These formulas encode the presence of an element in $Q_1 \times \Sigma$ (a state in the first semiautomaton $\mathcal{A}$ and an alphabet symbol) for the half-reset $\mathcal{H}$.

As $\mathcal{H}$ is a half-reset, we can classify every tuple $(q, w) \in Q_1 \times \Sigma$ into one of two sets Σ_0 or Σ_1 (depending on if they reset to q_0 or q_1 in $\mathcal{H}$). As in Lem. C.1, we can define predicates $\mathcal{B}_{\mathcal{H},q}(t)$ using the predicates $Q'_{(q,w)}$ and the known classification of elements in $Q_1 \times \Sigma$ into some state q_0 or q_1.

To finally simulate the cascade product $\mathcal{C} = \mathcal{A} \circ \mathcal{H}$, we define predicates that compute for every state (q, p), $q \in Q_1$, $p \in Q_2$, whether $\mathcal{C}$ is in it:

$$\mathcal{C}_{(q,p)}(t) = \mathcal{B}_{\mathcal{A},q}(t) \wedge \mathcal{B}_{\mathcal{H},p}(t) \tag{25}$$

■

Theorem C.1. *Let $\mathcal{A}$ be a POFA. Then, there exists an equivalent transformer $\mathsf{T} \in \mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$.*

Proof. Let $\mathcal{C} = \mathcal{B}_0 \circ \cdots \circ \mathcal{B}_k$ the semiautomaton $\mathcal{A}$ is homomorphic to. Let $\phi\colon Q' \to Q$ be the homomorphism from $\mathcal{C}$ to $\mathcal{A}$ (where Q are the states of $\mathcal{A}$, Q' are the states of $\mathcal{C}$). By Lem. C.2, we can iteratively define formulas $\mathcal{C}_{q'}$ for all $q' \in Q'$ that simulate the semiautomaton $\mathcal{C}$. If we write instructions describing the homomorphism ϕ, we can then write formulas that yield the states traversed by $\mathcal{A}$ as:

$$\mathcal{A}_q(t) = \bigvee_{p \in Q', \phi(p)=q} \mathcal{C}_p(t) \tag{26}$$

$\mathcal{A}_q(t)$, however, describes the state *before reading the symbol at* t (by strict masking). We want predicates that yield the state *after reading at the symbol position* t. We thus write:

$$Y_q(t) = \bigvee_{\substack{p \in Q, w \in \Sigma \\ \delta(p,w)=q}} \mathcal{A}_p(t) \wedge Q_w(t) \tag{27}$$

Finally, we denote by F the set of final states in $\mathcal{A}$. We thus define the output vector Y by:

$$Y(t) = \bigvee_{q \in F} Y_q(t) \tag{28}$$

$Y(T) = 1$ if and only if $\mathcal{A}$ is in one of the final states when reading the entire string. This concludes the translation of $\mathcal{A}$ to **B-RASP**$^{\mathsf{F}}_{\blacktriangleleft}$, showing the existence of a $\mathsf{T} \in \mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$ equivalent to $\mathcal{A}$. ■

D Direct Translations of $\mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$

D.1 A Normal Form for $\mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$ Transformers

The representational capacity of transformers depends tightly on the modeling assumptions. §6 studies the same architecture as Yang et al. (2024a). In this section, we outline a formalization of $\mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$ transformers in a form that allows us to describe direct translations to **LTL**$[\diamondminus]$ and POFAs in §7 more easily. The idea of this is similar to the normal form of Hao et al. (2022).

D.1.1 Finite Precision and Simplifications

As Yang et al. (2024a), we work with finite-precision transformers with no positional encodings. Further, we focus on strict future masking and leftmost hard attention, defined formally below.

In our formalization, we omit element-wise transformations such as the query, key, and value transformations and element-wise MLPs. Instead, we directly aggregate the original representations into (constant-size) vectors that can take increasingly many values as the number of layers increases.[7] The intuition and motivation for this stems from the use of hard attention: At any layer, hard attention augments the current contextual representation with the representation of one (preceding) symbol—the $\mathrm{hardmax}$. Thus, all information that the transformer can capture is already captured if the contextual representations keep around the identities of the elements that were returned by $\mathrm{hardmax}$. The element-wise representations, in this case, do not provide any additional representational power. We elaborate on this in App. D.1.3.

Note that we do not use layer normalization in our transformers and focus on a single-head attention mechanism. Both can be incorporated into our proofs analogously to the solutions presented in Yang et al. (2024a) (see Yang et al. (2024a, §4.1) for a discussion of multi-head attention and Yang et al. (2024a, §4.3) for layer normalization).

D.1.2 The Attention Mechanism

The central component of a transformer is the **transformer layer**.

Definition D.1. *A **transformer layer** $\mathcal{L}\colon (\mathbb{R}^D)^+ \to (\mathbb{R}^D)^+$ is a length-preserving function defined as*

$$\mathcal{L}((\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)) \stackrel{\text{def}}{=} (\boldsymbol{y}_t)_{t=1}^T \tag{29a}$$

$$(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T) \stackrel{\text{def}}{=} \mathtt{att}((\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)) + (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$$

[7] This is formalized in Lem. D.1.

We will use $\mathcal{L}_t \stackrel{\text{def}}{=} \texttt{proj}_t \circ \mathcal{L}$ for the function that computes $\mathcal{L}((\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T))$ and extracts the contextual representation of the tth symbol by projecting out that dimension.

The **attention mechanism** att is specified with the following components:

- A **scoring function** score: $\mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$.
- A **masking function** $m\colon \mathbb{N} \times \mathbb{N} \to \{0, 1\}$ that determines the positions attended to. We use future and past masking as in **B-RASP**. We write $\mathcal{M}(t) \stackrel{\text{def}}{=} \{t' \mid m(t, t') = 1\}$.
- A **normalization function** norm: $\mathbb{R}^T \to \boldsymbol{\Delta}^{T-1}$ that normalizes the attention scores.

We then define the attention mechanism as:

$$\texttt{att}((\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)) \stackrel{\text{def}}{=} (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T) \tag{30a}$$

$$\boldsymbol{y}_t \stackrel{\text{def}}{=} \sum_{t' \in \mathcal{M}(t)} s_{t'} \boldsymbol{x}_{t'} \tag{30b}$$

$$\boldsymbol{s} \stackrel{\text{def}}{=} \texttt{norm}((\texttt{score}(\boldsymbol{x}_{t'}, \boldsymbol{x}_t))_{t' \in \mathcal{M}(t)}) \tag{30c}$$

The $\sum$-notation in Eq. (30b) can naturally be thought of as *collecting* information from the unmasked positions. Intuitively, if the space of contextual representations is large enough, this can be interpreted as *concatenating* the representations together. This will be particularly useful in our UHA formulation, where only a single $\boldsymbol{x}_{t'}$ will be selected and the finiteness of the representation space will mean that *all* relevant information about the string $\boldsymbol{w}_{\leq t}$ will be stored in $\boldsymbol{x}_t$. See also App. D.1.3.

Let $L \in \mathbb{N}_{\geq 1}$ and $\mathcal{L}_1, \ldots, \mathcal{L}_L$ be transformer layers. A **transformer** $\mathsf{T}\colon \Sigma^* \to (\mathbb{R}^D)^+$ is a composition of transformer layers:

$$\mathsf{T} \stackrel{\text{def}}{=} \mathcal{L}_L \circ \cdots \circ \mathcal{L}_1 \circ \texttt{embed} \tag{31}$$

where $\texttt{embed}\colon \Sigma^* \to \left(\mathbb{R}^D\right)^+$ is a position-wise embedding function that maps symbols to their static representations.

We also write

$$(\boldsymbol{x}_1^{(\ell)}, \ldots, \boldsymbol{x}_T^{(\ell)}) = (\mathcal{L}_\ell \circ \cdots \circ \mathcal{L}_1 \circ \texttt{embed})(\boldsymbol{w}) \tag{32}$$

for some layer index $\ell \in [L]$ and string $\boldsymbol{w} = w_1 \cdots w_T \in \Sigma^*$. We call $\boldsymbol{x}_t^{(\ell)}$ the **contextual representation** of symbol w_t at layer ℓ.

A transformer computes the contextual representations of the string $\boldsymbol{w} = w_1 \cdots w_T \text{EOS}$ as

$$(\boldsymbol{x}_1^{(L)}, \ldots, \boldsymbol{x}_T^{(L)}, \boldsymbol{x}_{\text{EOS}}^{(L)}) \stackrel{\text{def}}{=} \mathsf{T}(\boldsymbol{w}). \tag{33}$$

We take $\boldsymbol{x}_{\text{EOS}}^{(\ell)}$ to be the representation of the entire string. This motivates the definition of the **transformer encoding function** $\boldsymbol{h}$:

$$\boldsymbol{h}(\boldsymbol{w}) \stackrel{\text{def}}{=} \boldsymbol{x}_{\text{EOS}}^{(L)}. \tag{34}$$

This allows us to define a transformer's language. This is usually defined based on a linear classifier based on $\boldsymbol{h}(\boldsymbol{w})$;

$$\mathbb{L}(\mathsf{T}) \stackrel{\text{def}}{=} \{\boldsymbol{w} \in \Sigma^* \mid \boldsymbol{\theta}^\top \boldsymbol{h}(\boldsymbol{w}) > 0\} \tag{35}$$

for some $\boldsymbol{\theta} \in \mathbb{R}^D$. Since we are working with finite-precision transformers, the set of possible $\boldsymbol{h}(\boldsymbol{w})$ is finite (in our normal form, it is a subset of Σ^{2^L}). We can thus equate the condition $\boldsymbol{\theta}^\top \boldsymbol{h}(\boldsymbol{w}) > 0$ with $\boldsymbol{h}(\boldsymbol{w})$'s inclusion in a subset of Σ^{2^L} and define the language of a transformer as follows.

Definition D.2. *Let $\mathsf{T} \in \mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$. We define its language $\mathbb{L}(\mathsf{T})$ as*

$$\mathbb{L}(\mathsf{T}) \stackrel{\text{def}}{=} \{\boldsymbol{w} \in \Sigma^* \mid \boldsymbol{h}(\boldsymbol{w}) \in \mathcal{F}_{\mathsf{T}}\} \tag{36}$$

where $\mathcal{F}_{\mathsf{T}}$ is a set of accepting final representations.

D.1.3 Unique Hard Attention

Let $\mathcal{C} \in \{\min, \max\}$ and let $\boldsymbol{s} \in \mathbb{R}^N$. We define

$$\mathrm{hardmax}(\boldsymbol{s})_n \stackrel{\text{def}}{=} \begin{cases} 1 & \textbf{if } n = \mathcal{C}(\mathrm{argmax}(\boldsymbol{s})) \\ 0 & \textbf{otherwise} \end{cases} \tag{37}$$

Here, $\mathrm{argmax}\, \boldsymbol{s}$ denotes the set of indices attaining the maximum value in the vector $\boldsymbol{s}$. The function $\mathcal{C}$ selects a unique index from this set: $\mathcal{C} = \max$ corresponds to **rightmost tiebreaking** ▶ and $\mathcal{C} = \min$ corresponds to **leftmost tiebreaking** ◀.

Definition D.3. ***Unique hard attention*** *is computed with the* $\mathrm{hardmax}$ *projection function:*

$$\mathrm{hardmax}(\boldsymbol{s}) = \mathrm{hardmax}(\boldsymbol{s}) \tag{38}$$

With some abuse of notation, we sometimes write $\mathrm{hardmax}(\boldsymbol{s})$ for the *position* $\mathcal{C}(\mathrm{argmax}(\boldsymbol{s}))$.

We denote future or past masking with F or P, respectively. $\mathcal{T}_{◀}^{\mathsf{F}}$, for example, denotes the class of transformers with future masking and leftmost attention.

The following lemma is a restatement of Yang et al. (2024a, Lem. 22).

Lemma D.1. *Let T be a UHA transformer over Σ. Denote with $\boldsymbol{x}_t^{(\ell)}$ the contextual representation of the symbol w_t at layer ℓ. The following holds:*

$$|\{\boldsymbol{x}_t^{(\ell)} \mid \boldsymbol{w} \in \Sigma^*, t \in [|\boldsymbol{w}|]\}| \leq |\Sigma|^{2^\ell}. \tag{39}$$

Proof. We prove the statement by induction on the number of layers.

Base case: $\ell = 1$. In the first layer, as we have static representations for symbols, the embedding at some position t is uniquely determined by the symbol w_t at that position. We thus have exactly $|\Sigma|$ possible representations for a given position, regardless of the length of the string.

Inductive step: $\ell > 1$. Suppose that the invariance holds for $\ell - 1$: $|\{\boldsymbol{x}_t^{(\ell-1)} \mid \boldsymbol{w} \in \Sigma^*, t \in [|\boldsymbol{w}|]\}| \leq |\Sigma|^{2^{\ell-1}}$. For any position t in the string, at layer ℓ, we will compute $\boldsymbol{x}_t^{(\ell)} = \mathtt{att}((\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)) + (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$ where $\boldsymbol{x}_t^{(\ell-1)}$ is the representation of the symbol at the previous layer $\ell - 1$. By the induction hypothesis, the element $\boldsymbol{x}_t^{(\ell-1)}$ takes one out of at most $|\Sigma|^{2^{\ell-1}}$ possible values. Moreover, the attention mechanism selects one element from the previous layer $\ell - 1$, which holds one out of at most $|\Sigma|^{2^{\ell-1}}$ possibles values by the induction hypothesis. The element $\boldsymbol{a}_t$ holds thus one out of $|\Sigma|^{2^{\ell-1}} \times |\Sigma|^{2^{\ell-1}} = |\Sigma|^{2^\ell}$ representations, concluding the induction step and the proof. ■

Contextual representations as elements of a finite set. Lem. D.1 allows us to simplify notation: Any representation of a symbol at layer ℓ is uniquely identified by 2^ℓ symbols, i.e., $\boldsymbol{x}^{(\ell)} \in \Sigma^{2^\ell}$. We think of this as the collection of representations attended to at each of layer $\ell' < \ell$, since each selects a single position to be added to the current representation. We will thus refer to $\boldsymbol{x}^{(\ell)}$ as elements of Σ^{2^ℓ}.

D.1.4 An Invariance

In this section and in App. D.2, we use Σ for the alphabet of input symbols and Ξ for a general alphabet (finite set) of relevance. Later, Ξ will correspond to sets of the form Σ^{2^ℓ} for some $\ell \in \mathbb{N}$.

Definition D.4. *Let Ξ be an alphabet and $\boldsymbol{w} \in \Xi^*$. We define the* ***symbol order*** *$\omega(\boldsymbol{w})$ of $\boldsymbol{w}$ as the string obtained from $\boldsymbol{w}$ by keeping only the first occurrence of each symbol.*

We define the following relation on Ξ^*:

$$\boldsymbol{w} \simeq_\omega \boldsymbol{w}' \iff \omega(\boldsymbol{w}) = \omega(\boldsymbol{w}'). \tag{40}$$

It is not hard to verify that $\simeq_\omega$ is an equivalence relation on Ξ^* and to verify that $|\Xi^* / \simeq_\omega| = |\mathrm{OS}(\Xi)|$, where $\mathrm{OS}(\Xi)$ is the (finite) set of all ordered subsets of Ξ. We denote the equivalence class of $\boldsymbol{w} \in \Xi^*$ by $[\boldsymbol{w}] \in \Xi^* / \simeq_\omega$. We have the following important invariance.

Lemma D.2 (Attention invariance)**.** *Let $\mathcal{L}$ be an $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ transformer layer over Ξ. For any $\boldsymbol{w}, \boldsymbol{w}' \in \Xi^*$, if $\boldsymbol{w} \simeq_\omega \boldsymbol{w}'$, then $\mathcal{L}_{|\boldsymbol{w}|}(\boldsymbol{w}) = \mathcal{L}_{|\boldsymbol{w}'|}(\boldsymbol{w}')$.*

Proof. This follows directly from the definition of leftmost hard attention: Additional occurrences of symbols to the right of the first occurrence do not change the position attended to, meaning that the output at the final symbol is the same. ■

D.2 $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ as LTL$[\diamondminus]$

We view a transformer layer as a function that takes a contextual representation $\boldsymbol{x}_t^{(\ell)}$ and returns a function that takes in an ordered set of contextual representations $\omega(\boldsymbol{w})$ with $\boldsymbol{w} \in \Xi^*$ and returns the representation chosen by the unique hard attention mechanism.

$$\mathcal{L}^{(\ell)} \colon \Xi \to \mathrm{Map}(\mathrm{OS}(\Xi), \Xi), \tag{41a}$$

$$\mathcal{L}^{(\ell)} \colon \boldsymbol{x} \mapsto \boldsymbol{\lambda}_{\boldsymbol{x}}. \tag{41b}$$

$\boldsymbol{\lambda}_{\boldsymbol{x}}$ is the function that takes an ordered set of contextual representations $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and returns the representation chosen by UHA:[8]

$$\boldsymbol{\lambda}_{\boldsymbol{x}} \colon \mathrm{OS}(\Xi) \to \Xi, \tag{42a}$$

$$\boldsymbol{\lambda}_{\boldsymbol{x}} \colon \mathcal{X} \mapsto \boldsymbol{x}_{\mathrm{hardmax}((\mathtt{score}(\boldsymbol{x}_t, \boldsymbol{x}_{t'}))_{\boldsymbol{x}_{t'} \in \mathcal{X}})}. \tag{42b}$$

We also define

$$\boldsymbol{x}'' \preceq_{\boldsymbol{x}} \boldsymbol{x}' \iff \mathtt{score}\big(\boldsymbol{x}, \boldsymbol{x}''\big) \leq \mathtt{score}\big(\boldsymbol{x}, \boldsymbol{x}'\big) \tag{43a}$$

$$\boldsymbol{x}'' \simeq_{\boldsymbol{x}} \boldsymbol{x}' \iff \mathtt{score}\big(\boldsymbol{x}, \boldsymbol{x}''\big) = \mathtt{score}\big(\boldsymbol{x}, \boldsymbol{x}'\big) \tag{43b}$$

$$\boldsymbol{x}'' \prec_{\boldsymbol{x}} \boldsymbol{x}' \iff \boldsymbol{x}'' \preceq_{\boldsymbol{x}} \boldsymbol{x}' \text{ and } \neg\left(\boldsymbol{x}'' \simeq_{\boldsymbol{x}} \boldsymbol{x}'\right). \tag{43c}$$

Theorem 7.1. *Let $\mathsf{T} \in \mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ be a transformer. Then, there exists an equivalent formula $\psi \in$ LTL$[\diamondminus]$.*

Proof. We define an LTL$[\diamondminus]$ formula representing $\mathsf{T} \in \mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ with the attention mechanism implemented by the function $\boldsymbol{\lambda}_{\boldsymbol{x}}$ (cf. Eq. (42b)) by specifying a set of formulas representing each layer, starting with the initial one, and continuing inductively. At a high level, we define the formulas $\phi_{\boldsymbol{x} \leftarrow \boldsymbol{y}}^{(\ell)}$ to mean that the contextual representation $\boldsymbol{y}$ is added to the contextual representation $\boldsymbol{x}$ at layer ℓ, i.e., that $\boldsymbol{y}$ is the maximizer of the query-key score when $\boldsymbol{x}$ is the query: $\boldsymbol{\lambda}_{\boldsymbol{x}}(\mathcal{X}) = \boldsymbol{y}$ for $\mathcal{X}$ representing the current string. We construct the formulas layer by layer.

Base case: $\ell = 1$. We begin by specifying first-layer formulas, which work over Σ. We define for $a, b \in \Sigma$:

$$\phi_{a \leftarrow b}^{(1)} \stackrel{\text{def}}{=} \pi_a \wedge \underbrace{\bigwedge_{\substack{w \in \Sigma: \\ b \prec_a w}} \neg\diamondminus\pi_w}_{\substack{\text{No previous symbols with} \\ \text{higher scores than b}}} \wedge \underbrace{\diamondminus\Big(\pi_b \wedge \bigwedge_{w \in \Sigma \setminus \{b\}: b \simeq_a w} \neg\diamondminus\pi_w\Big)}_{\substack{\text{There is } b \text{ in the past with no} \\ \text{equally-scored symbols in its past}}}. \tag{44}$$

In words, b's value is added to a's static embedding if *(i)* there are no symbols in the past of a that have a higher score than b and *(ii)* there exists a position with b in the past such that there are no symbols with equal scores to its left (leftmost tiebreaking).

[8] We omit the dependence of $\boldsymbol{\lambda}$ on the layer index for readability.

Inductive step: $\ell > 1$. Let now $\mathcal{L}$ be the transformer layer at layer $\ell > 1$ and assume we have correctly constructed $\phi^{(\ell')}_{\boldsymbol{x} \leftarrow \boldsymbol{y}}$ for $\ell' < \ell$.

Firstly, we define the formulas $\pi^{(\ell)}_{\boldsymbol{x}}$ that, analogously to π, specify the presence of contextual representations for elements in Σ^{2^ℓ}. Writing $\boldsymbol{x} \in \Sigma^{2^\ell}$ as $\boldsymbol{x} = (\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{\ell-1})$ with $\boldsymbol{z}_0 \in \Sigma$ and $\boldsymbol{z}_{\ell'} \in \Sigma^{2^{\ell'}}, \ell' \in \{0, \ldots, \ell-1\}$, we define:

$$\pi^{(\ell)}_{\boldsymbol{x}} = \pi_{\boldsymbol{z}_0} \wedge \underbrace{\bigwedge_{\ell'=1}^{\ell-1} \phi^{(\ell')}_{\boldsymbol{x}_{\leq \ell'-1} \leftarrow \boldsymbol{z}_{\ell'}}}_{\substack{\text{Verify correct representations} \\ \text{throughout layers are added to } \boldsymbol{z}_0 \in \Sigma}} \tag{45}$$

where $\boldsymbol{x}_{\leq \ell'} = (\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{\ell'})$. This checks the presence of $\boldsymbol{x} \in \Sigma^{2^\ell}$ as the contextual representation by asserting that the individual representations in $\boldsymbol{x}$ at lower levels were indeed added by checking the formulas $\phi^{(\ell')}_{\boldsymbol{x}_{\leq \ell'-1} \leftarrow \boldsymbol{z}_{\ell'}}$.

We now define, for $\boldsymbol{x}, \boldsymbol{y} \in \Xi = \Sigma^{2^\ell}$:

$$\phi^{(\ell)}_{\boldsymbol{x} \leftarrow \boldsymbol{y}} = \bigvee_{\boldsymbol{\mathcal{X}} \in \mathrm{OS}(\Xi)} \Bigg[\underbrace{\mathsf{order}^{(\ell-1)}(\boldsymbol{\mathcal{X}})}_{\substack{\text{Identify correct ordered subset of} \\ \text{representations in the previous layer}}} \wedge \underbrace{\mathsf{best}^{(\ell-1)}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mathcal{X}})}_{\substack{\text{Check if } \boldsymbol{y} \text{ is best representation for } \boldsymbol{x} \\ \text{given the correct ordered subset } \boldsymbol{\mathcal{X}}}} \Bigg] \tag{46}$$

Intuitively, the formula iterates over all possible ordered subsets of Ξ, checks which one describes the string in the past, and then asserts whether $\boldsymbol{y}$ is the best symbol to add to $\boldsymbol{x}$ given the set of contextual representations $\boldsymbol{\mathcal{X}}$. Here, $\mathsf{order}^{(\ell)}$ is a formula that checks whether $\omega(\boldsymbol{w}) = \boldsymbol{\mathcal{X}}$ by making sure that the string in the past follows the same order as $\boldsymbol{\mathcal{X}}$:

$$\mathsf{order}^{(\ell)}(\boldsymbol{\mathcal{X}}) \stackrel{\text{def}}{=} \underbrace{\left[\Diamond(\pi^{(\ell)}_{\boldsymbol{z}_1} \wedge \Diamond(\pi^{(\ell)}_{\boldsymbol{z}_2} \wedge \cdots \wedge (\Diamond \pi^{(\ell)}_{\boldsymbol{z}_{|\boldsymbol{\mathcal{X}}|}})))\right]}_{\text{Elements of } \boldsymbol{\mathcal{X}} \text{ are present in correct order}} \wedge \underbrace{\bigwedge_{\boldsymbol{z} \in \Xi \setminus \boldsymbol{\mathcal{X}}} \neg \Diamond \pi^{(\ell)}_{\boldsymbol{z}}}_{\substack{\text{Representations not in } \boldsymbol{\mathcal{X}} \\ \text{are not present}}} \tag{47}$$

Analogously to Eq. (44), $\mathsf{best}^{(\ell)}$ checks whether $\boldsymbol{y}$ is the best symbol to add to $\boldsymbol{x}$ given the set of contextual representations $\boldsymbol{\mathcal{X}}$ by *(i)* asserting $\boldsymbol{x}$ is at the current position, *(ii)* there are no representations in the past of $\boldsymbol{x}$ given the current ordered subset $\boldsymbol{\mathcal{X}}$ that have a higher score than $\boldsymbol{y}$ and *(iii)* there exists a position in $\boldsymbol{\mathcal{X}}$ with $\boldsymbol{y}$ in the past such that there are no representations with equal scores to its left (leftmost tiebreaking).

$$\mathsf{best}^{(\ell)}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mathcal{X}}) \stackrel{\text{def}}{=} \pi^{(\ell)}_{\boldsymbol{x}} \wedge \underbrace{\bigwedge_{\substack{\boldsymbol{z} \in \boldsymbol{\mathcal{X}}: \\ \boldsymbol{y} \prec_{\boldsymbol{x}} \boldsymbol{z}}} \neg \Diamond \pi^{(\ell)}_{\boldsymbol{z}}}_{\substack{\text{No previous representations} \\ \text{with higher scores than } \boldsymbol{y}}} \wedge \Diamond \Bigg(\underbrace{\pi^{(\ell)}_{\boldsymbol{y}} \wedge \bigwedge_{\substack{\boldsymbol{z} \in \boldsymbol{\mathcal{X}} \setminus \{\boldsymbol{y}\}: \\ \boldsymbol{y} \simeq_{\boldsymbol{x}} \boldsymbol{z}}} \neg \Diamond \pi^{(\ell)}_{\boldsymbol{z}}}_{\substack{\text{There is } \boldsymbol{y} \text{ in the past with no} \\ \text{equally-scored representations in its past}}} \Bigg). \tag{48}$$

Finally, let $\mathcal{L}$ be the final transformer layer and let $F \subseteq \Sigma^{2^L}$ be the set of representations for EOS that lead to string acceptance by T. The **LTL**$[\Diamond]$ formula ψ representing T simply has to check whether the representation for EOS is in F:

$$\psi = \bigvee_{\boldsymbol{x} \in F} \pi^{(L)}_{\boldsymbol{x}}. \tag{49}$$

■

D.3 $\mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$ as POFAs

Theorem D.1. *Let $\mathsf{T} \in \mathcal{T}^{\mathsf{F}}_{\blacktriangleleft}$ be a transformer. Then, there exists an equivalent POFA.*

Proof. We will construct a semiautomaton $\mathcal{A}$ that will, after reading $\boldsymbol{w}_{\leq t}$, store in its state the ordered subsets $\mathcal{X}^{(\ell)} \in \mathrm{OS}(\Sigma^{2^\ell})$ of contextual representations for all the transformer layers. In other words, it will hold L *equivalence classes* of the string $\boldsymbol{w}_{\leq t}$, one for each layer. This state will be updated sequentially according to the self-attention mechanism implemented by the transformer.

Formally, given a transformer $\mathsf{T} \in \mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ with the attention mechanism implemented by the function $\boldsymbol{\lambda}_{\boldsymbol{x}}$ (cf. Eq. (42b)) over the alphabet Σ, we define the semiautomaton $\mathcal{A} = (\Sigma, Q, \delta)$. We take the set of states Q to be

$$Q \stackrel{\text{def}}{=} \underbrace{\mathrm{OS}(\Sigma) \times \cdots \times \mathrm{OS}(\Sigma^{2^L})}_{\text{Ordered sets of representations of all layers.}} . \tag{50}$$

For clarity, we will explicitly write out the states $q \in Q$ in their components:

$$q = \begin{pmatrix} \mathcal{X}^{(0)} \\ \mathcal{X}^{(1)} \\ \vdots \\ \mathcal{X}^{(L)} \end{pmatrix} \tag{51}$$

with $\mathcal{X}^{(\ell)} \in \mathrm{OS}(\Sigma^{2^\ell})$ for $\ell \in \{0, \ldots, L\}$.

$\mathcal{A}$ will update $\mathcal{X}^{(\ell)}$ with new occurrences of contextual representations by "appending" new contextual representations. Let us describe how the transition function δ updates the state q upon reading the symbol w. We write q for the source state and $\mathcal{X}^{(\ell)}$ for its components, and q' for the target state and $\mathcal{X}'^{(\ell)}$ for its components.

We then define $\boldsymbol{x}'^{(0)} = w$ to mean that the static representation of this symbol is the symbol itself. For $\ell \geq 1$, we define

$$\boldsymbol{x}'^{(\ell)} \stackrel{\text{def}}{=} \begin{pmatrix} \boldsymbol{x}'^{(\ell-1)} \\ \boldsymbol{\lambda}_{\boldsymbol{x}'^{(\ell-1)}}\left(\mathcal{X}^{(\ell-1)}\right) \end{pmatrix} \tag{52}$$

which simulates the ℓth layer of T by *(1)* copying the symbol's representation $\boldsymbol{x}'^{(\ell-1)}$ from the previous layer into the first component (residual connection) and *(2)* computing the attended-to representation based on all the contextual representations seen so far at the previous layer ($\mathcal{X}^{(\ell-1)}$) and the symbol's representation at the previous layer ($\boldsymbol{x}'^{(\ell-1)}$). Crucially, Eq. (52) can be computed *in advance* (knowing the scoring function `score`) for any $\mathcal{X}^{(\ell)} \in \mathrm{OS}(\Sigma^{2^\ell})$ and $\boldsymbol{x} \in \Sigma^{2^\ell}$ for all $\ell \in \{0, \ldots, L\}$ due to the finiteness of all the considered states.

We then update the set of observed contextual representations as

$$\mathcal{X}'^{(\ell)} \stackrel{\text{def}}{=} \mathcal{X}^{(\ell)} \cup \{\boldsymbol{x}'^{(\ell)}\} \tag{53}$$

to incorporate the information about the new contextual representations into the ordered sets of seen contextual representations at each layer $\ell \in \{0, \ldots, L\}$. The union in Eq. (53) is to be interpreted as adding an element to an ordered set.

Defining

$$q' = \begin{pmatrix} \mathcal{X}'^{(0)} \\ \mathcal{X}'^{(1)} \\ \vdots \\ \mathcal{X}'^{(L)} \end{pmatrix} \tag{54}$$

and setting $\delta(q, w) = q'$, for all choices of q and w, we have defined the transition function δ that updates each state with the new contextual representations at each layer.

The set of observed contextual representations $\mathcal{X}$ satisfies a partial order as the concatenation of a new representation at a new position at every step implies we transition into novel states. This further implies that the semiautomaton is partially ordered.

We construct an *automaton* from $\mathcal{A}$ by setting the initial and final states. We set the initial state to be the one with empty ordered subsets of observed representations: $\mathcal{X}^{(\ell)} \stackrel{\text{def}}{=} \emptyset$ for all $\ell \in \{0, \ldots, L\}$. A subset of $\mathrm{OS}(\Sigma^{2^L})$ will lead to T accepting a string. We set the states with $\mathcal{X}^L$ in that set to be final, leading to an equivalent automaton. ■

E Duality with $\mathcal{T}_{\blacktriangleright}^{\mathsf{P}}$, $\mathbf{LTL}[\diamond\!\!\!\!\cdot]$

The class of transformers $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$, the main focus of the paper, has a natural symmetric characterization in $\mathcal{T}_{\blacktriangleright}^{\mathsf{P}}$: While $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ can only peek strictly into the past, $\mathcal{T}_{\blacktriangleright}^{\mathsf{P}}$ can symmetrically only peek strictly into the future using strict past masking and rightmost UHA. $\mathcal{T}_{\blacktriangleright}^{\mathsf{P}}$ can be informally seen as transformers in $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ that instead read symbols *right-to-left*, only considering the future when updating a symbol representation, analogously to the duality between $\mathcal{R}$-trivial and $\mathcal{L}$-trivial languages (Brzozowski and Ellen, 1980).

Thus, all results for $\mathcal{T}_{\blacktriangleleft}^{\mathsf{F}}$ and $\mathbf{LTL}[\diamond]$ apply analogously to the symmetric case of $\mathcal{T}_{\blacktriangleright}^{\mathsf{P}}$ and $\mathbf{LTL}[\diamond\!\!\!\!\cdot]$. $\mathbf{LTL}[\diamond\!\!\!\!\cdot]$ is the dual of $\mathbf{LTL}[\diamond]$—it only permits peeking into the future rather than the past. Similarly, partially ordered reverse automata (RPOFA) are semiautomata whose reversal (automaton constructed by inverting the directionality of the transitions) are POFAs. The reverse of a RPOFA is then homomorphic to a cascade product of half-resets. We thus can write the dual statement of Thm. 3.2.

Theorem E.1. *Let $T \in \mathcal{T}_{\blacktriangleright}^{\mathsf{P}}$ be a transformer. Then, there exists an equivalent formula $\psi \in \mathbf{LTL}[\diamond\!\!\!\!\cdot]$.*

Let $\psi \in \mathbf{LTL}[\diamond\!\!\!\!\cdot]$ be a formula. Then, there exists an equivalent transformer $T \in \mathcal{T}_{\blacktriangleright}^{\mathsf{P}}$.

Enhancing Input-Label Mapping in In-Context Learning with Contrastive Decoding

Keqin Peng[1], **Liang Ding**[2*], **Yuanxin Ouyang**[1], **Meng Fang**[3], **Yancheng Yuan**[4], **Dacheng Tao**[5]
[1]Beihang University [2]The University of Sydney [3]University of Liverpool
[4]The Hong Kong Polytechnic University [5]Nanyang Technological University
keqin.peng@buaa.edu.cn, liangding.liam@gmail.com

Abstract

Large language models (LLMs) excel at a range of tasks through in-context learning (ICL), where only a few task examples guide their predictions. However, prior research highlights that LLMs often overlook input-label mapping information in ICL, relying more on their pre-trained knowledge. To address this issue, we introduce In-Context Contrastive Decoding (ICCD), a novel method that emphasizes input-label mapping by contrasting the output distributions between positive and negative in-context examples. Experiments on 7 natural language understanding (NLU) tasks show that our ICCD method brings consistent and significant improvement (up to +1.8 improvement on average) upon 6 different scales of LLMs without requiring additional training. Our approach is versatile, enhancing performance with various demonstration selection methods, demonstrating its broad applicability and effectiveness. The code and scripts are released at https://github.com/Romainpkq/CD_ICL.

1 Introduction

In-context learning (ICL, Brown et al., 2020) is one of the most remarkable emergent capabilities of large language models (LLMs, Achiam et al., 2023; Dubey et al., 2024). By leveraging just a few carefully selected input-output examples, ICL enables models to adapt to new tasks without parameter updating (Dong et al., 2022; Peng et al., 2024). This approach has proven highly effective in unlocking the advanced capabilities of LLMs and has become a standard technique for tackling a spectrum of tasks, like translation, coding, and reasoning (Peng et al., 2023; Wang et al., 2025; Wibisono and Wang, 2024).

Previous studies (Pan et al., 2023; Wei et al., 2023) have identified two critical factors for successful ICL: *task recognition (TR)*, which involves identifying the task from the demonstrations and utilizing prior knowledge to make predictions, and *task learning (TL)*, which focuses on directly learning the input-label mappings from the demonstrations. However, ICL faces challenges in overcoming the biases introduced by pretraining (Kossen et al., 2024), and LLMs tend to underutilize input-label mapping information (Min et al., 2022). For example, in tasks like SST-2 (Socher et al., 2013b), the model may default to using its internal knowledge rather than learning the specific input-label mappings provided in the context.

To address this issue, we propose a simple yet effective method called *in-context contrastive decoding* (ICCD). Our method is inspired by the contrastive decoding technique (Li et al., 2023; Sennrich et al., 2024; Kim et al., 2024; Zhong et al., 2024; Wang et al., 2024), which increases the probability of the desired output by suppressing undesired outputs, and our ICCD enhances the model's attention to input-label mapping during generation. Specifically, we construct negative in-context examples by altering the inputs of the demonstrations, creating incorrect input-label mappings while keeping the labels unchanged. By comparing the output distributions between positive and negative examples, ICCD effectively emphasizes the correct input-label mappings, integrating this information into the original ICL process. Notably, our method works with any pretrained LLMs without requiring additional training.

Experimental results across seven natural language understanding tasks demonstrate that our ICCD strategy consistently and significantly improves performance upon several advanced LLMs, e.g., Llama-3.1, Llama-3.2, and Qwen2, across various datasets and model scales. Moreover, we show that ICCD can be seamlessly integrated with different demonstration selection methods, showcasing its robustness and universal applicability.

* Corresponding Authors.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 997–1004
July 27 - August 1, 2025

2 Methodology

2.1 Background

Given an input query x, the probability of generating the target y using a casual LLM M parameterized by θ can be formulated as follows:

$$y \sim p_\theta(y \mid \boldsymbol{c}, \mathcal{T}(x)), \tag{1}$$

where $\mathcal{T}(\cdot)$ is the template used to wrap up inputs and $c = \mathcal{T}(x_1), \cdots, \mathcal{T}(x_k)$ is the context string concatenating k in-context examples, $p_\theta(y \mid \boldsymbol{c}, \mathcal{T}(x)) = \text{softmax}[\text{ logit}_\theta(y \mid \boldsymbol{c}, \mathcal{T}(x))]$ is the probability for the predicted token. For obtaining the desired y, the regular decoding method is to choose the token with the highest probability (*i.e.*, greedy decoding) or sampling from its distribution (*e.g.*, top-k decoding).

Here, we can observe that there are two kinds of knowledge contributing to model prediction, models' prior knowledge and input-label mapping information in in-context learning. However, LLMs usually prioritize prior knowledge over input-label mapping information (Kossen et al., 2024), leading to ICL's struggle to fully overcome prediction preferences acquired from pre-training.

2.2 In-Context Contrastive Decoding

To mitigate the issue above, we construct negative in-context examples to factor out the input-label mapping from the models' original output distribution contrastively. Specifically, in addition to the origin in-context examples $\boldsymbol{c}$, we construct negative in-context examples $\boldsymbol{c}^-$ with incorrect input-label mapping. We then subtract the negative output $\mathbf{z}_t^-$ from the positive output $\mathbf{z}_t$ to isolate the knowledge of input-label mapping. Finally, we integrate this knowledge with the original in-context learning to reinforce the importance of input-label mapping:

$$y_t \sim \text{softmax}(\mathbf{z}_t + \alpha(\mathbf{z}_t - \mathbf{z}_t^-)), \tag{2}$$

where α is a hyperparameter that governs the importance of input-label mapping information. Equivalently,

$$y_t \sim \tilde{p}_\theta(y|\boldsymbol{c}, \boldsymbol{c}^-, \mathcal{T}(x)) \tag{3}$$

$$\propto p_\theta(y|\boldsymbol{c}, \mathcal{T}(x)) \left(\frac{p_\theta(y|\boldsymbol{c}, \mathcal{T}(x)}{p_\theta(y|\boldsymbol{c}^-, \mathcal{T}(x))} \right)^\alpha . \tag{4}$$

Construction of $\boldsymbol{c}^-$. The negative in-context examples $\boldsymbol{c}^-$ is the key to the success of the in-context contrastive decoding method (ICCD). Considering the label bias (Zhao et al., 2021) of in-context learning, directly altering the labels of demonstrations may introduce a completely different label bias, potentially distorting the input-label mapping information. Hence, we adjust the inputs instead of the labels to change input-label mapping information. Specifically, for each demonstration (x_i, y_i), we first randomly select a different label $y_j (y_j \neq y_i)$ from the label space. Then we randomly choose an input x_j whose label is y_j from the demonstrations pool to construct the negative demonstration (x_j, y_i). We compare the effect of different $\boldsymbol{c}^-$ in Section 5.

3 Experimental Setup

Models and Baselines. We perform experiments across different sizes of models, including Llama-series: Llama3.2-1B (1B), Llama3.2-3B (3B) and Llama3.1-8B (8B) (Dubey et al., 2024) and Qwen2 series: Qwen2-0.5B (0.5B), Qwen2-1.5B (1.5B) and Qwen2-7B (7B) (Yang et al., 2024), which are all widely-used decoder-only dense LMs. We also conduct experiments on extensive alignment models, e.g., Llama3.2-1B-Instruct, Llama3.2-3B-Instruct, and Llama3.1-8B-Instruct (Dubey et al., 2024) to verify the generalizability of our approach. For the baseline, we use the regular decoding methods following prior work (Shi et al., 2024; Zhao et al., 2024).

Demonstration Selection methods. To verify that our method is complementary to different demonstration selection methods, we mainly consider three different demonstration selection methods that do not require additional training.

- **Random** baseline randomly select in context examples for each testing sample.
- **BM25** (Robertson et al., 2009) baseline uses BM25 to calculate the word-overlap similarity between samples and test input and select the high-similarity samples as demonstrations.
- **TopK** (Liu et al., 2022) baseline uses the nearest neighbors of a given test sample as the corresponding in-context examples.

Datasets and Metrics. We conduct a systematic study across 7 NLU tasks, including binary, multi-class classification tasks (SST-2, SST-5 (Socher et al., 2013a), CR, Subj (Wang et al., 2018)) and natural language inference tasks: MNLI (Williams

et al., 2018) and QNLI (Wang et al., 2018). We will report the accuracy to show the performance.

Experimental Details. Our method introduces a hyperparameter α to control the input-label mapping information. For simplicity, we set $\alpha = 1$ for all models and settings. We ran all experiments 3 times with different random seeds and reported the average accuracies. We use 16-shot ICL for all models. Without a special statement, we report the results of the random selection method.

4 Main Results

We demonstrate the effectiveness of our method in 7 NLU tasks described in the Datasets and Metrics section. We summarize the results in Table 1, Table 2, Tabel 3, and Figure 1. Based on the results, we can find that:

Our method brings gain across different tasks and model scales. Results on Table 2 show that our method can achieve consistently better performance across the majority of tasks under different model scales than the regular decoding method. Specifically, our method brings over 1.0 improvements (in accuracy) in all Llama-series models and Qwen2-series models. It's worth highlighting that ICCD brings +2.3 gains on average in the Qwen2-1.5B model. Furthermore, it is noteworthy that our approach can achieve more significant improvements in challenging tasks with the increase of model scale, such as QNLI and MNLI tasks, respectively bringing 5.1% (1.4%) and 1.8% (1.2%) gains compared to regular decoding in Llama3.1-8B (Qwen2-7B), demonstrating the effectiveness and universality of our method.

Our method consistently improves the performance with different in-context examples selection methods. Table 1 lists the average performance and standard deviation of different models with different demonstration selection methods. Clearly, our method can achieve better and stable performance with different demonstration selection methods. When the model scale increases, our method can achieve more improvement gains compared to the regular decoding method, +0.5 and +1.1 with BM25 method under Llama3.2-3B and Llama3.1-8B, respectively. These results prove that ICCD can be complementary with different demonstration selection methods.

Model	Decoding	Random		BM25		TopK	
		avg.	std.	avg.	std.	avg.	std.
Llama3.2-1B	Regular	66.1	-	72.5	-	73.6	-
	Ours	68.3	0.19	72.9	0.11	73.4	0.17
Llama3.2-3B	Regular	72.9	-	76.6	-	76.7	-
	Ours	74.6	0.47	77.1	0.28	76.9	0.19
Llama3.1-8B	Regular	77.6	-	79.7	-	80.2	-
	Ours	79.4	0.19	80.8	0.15	80.9	0.05

Table 1: **Average performance and standard deviation of 7 Natural Language Understanding (NLU) tasks with different in-context example selection methods. Red** results indicate that our method brings improvement over the regular decoding, while **Green** results denote no improvement.

Our method works for aligned chat models. To verify the effectiveness of our method for the chat LLMs, we conducted experiments on different instruction-tuned and RLHF-tuned LLMs. Figure 1 show that our method can achieve consistent improvement in different chat models, demonstrating that our method also works for instruction-tuned and safety-enhanced models.

Our method works for a larger number of target classes. To verify the effectiveness of our method for a larger number of target classes, we conducted experiments on datasets TREC (6 target classes) and Dbpedia (14 target classes) with the random selection method. Results on Table 3 show that our method can achieve remarkable improvement, demonstrating the effectiveness of our method in larger target classes.

5 Analysis

To further explore the impact of different factors on the effectiveness of our method, we conduct further analysis with the Llama3.2-8B models.

Effects of Different Negative In-context Examples. As mentioned in Section 2.2, the choice of negative in-context examples is important to the performance of our methods. Here, we conduct contrastive experiments to analyze the impact of different negative examples. Specifically, we refer to the selected negative examples as **Input**, if the input-label mapping is altered by modifying the inputs of the demonstrations. Additionally, we construct another variant, **Label**, in which the labels of the demonstrations are changed. For comparison, we also include **NULL**, which does not use any negative demonstrations, similar to Shi et al. (2024). The results in Table 4 show that **Input** out-

Model	Decoding	SST2	CR	SST5	Subj	QNLI	MNLI	AG_NEWS	*Avg.*
Llama3.2-1B	Regular	89.8	83.0	43.7	72.8	53.5	36.6	83.3	66.1
	Ours	**91.1**	**83.7**	**43.3**	**83.0**	**53.8**	**39.2**	**84.1**	**68.3 (+2.1)**
Llama3.2-3B	Regular	93.7	87.2	46.2	86.0	54.2	56.9	86.4	72.9
	Ours	**94.0**	**88.1**	**46.5**	**92.1**	**57.2**	**57.0**	**86.9**	**74.6 (+1.7)**
Llama3.1-8B	Regular	96.7	92.3	48.0	94.0	60.3	65.3	86.7	77.6
	Ours	**96.5**	**93.2**	**49.3**	**96.1**	**65.4**	**67.5**	**87.6**	**79.4 (+1.8)**
Qwen2-0.5B	Regular	87.9	89.4	34.5	62.2	52.5	47.6	78.1	64.6
	Ours	**89.2**	**89.6**	**33.9**	**68.1**	**53.2**	**47.6**	**78.7**	**65.8 (+1.2)**
Qwen2-1.5B	Regular	95.2	91.0	49.0	72.3	60.2	61.8	76.7	72.3
	Ours	**95.1**	**91.3**	**48.3**	**81.5**	**61.8**	**65.2**	**79.1**	**74.6 (+2.3)**
Qwen2-7B	Regular	96.0	91.5	51.9	82.3	71.4	78.7	83.8	79.4
	Ours	**96.3**	**91.7**	**52.9**	**90.4**	**72.8**	**79.9**	**85.0**	**81.3 (+1.9)**

Table 2: **Performance of different models across 7 Natural Language Understanding (NLU) tasks.** **Red** results indicate our method brings improvement over the regular decoding, while **Green** denote no improvement.

Model	Decoding	TREC	Dbpedia
Llama3.2-1B	**Regular**	40.0	85.6
	Ours	**46.2 (+6.2)**	**90.5 (+4.9)**
Llama3.2-3B	**Regular**	44.4	83.1
	Ours	**49.6 (+5.2)**	**91.4 (+8.3)**
Llama3.1-8B	**Regular**	41.0	87.5
	Ours	**46.6 (+5.6)**	**93.8 (6.3)**

Table 3: **Average performance of two datasets with larger target classes.** **Red** results indicate that our method brings improvement over the regular decoding.

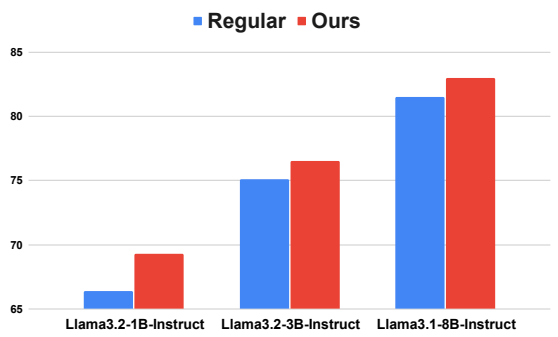


Figure 1: **Performance with different chat models.**

performs the other counterparts, thus leaving it as our default setting in this work.

Differences between the positive and negative examples. To verify whether our proposed method can truly lead to models to contrast the positive and negative in-context examples, we calculate the average KL divergence between the output distributions and report the results in Table 5, we can notice that our method can get large KL divergence in most datasets, which means that the output distributions of positive and negative in-context examples are notably different. This demonstrates that our method can truly lead to models to contrast the positive and negative in-context examples.

Method	Selection Method		
	Random	**BM25**	**TopK**
Regular Decoding	77.6	79.7	80.2
Equipped with our method			
+NULL	**73.0**	**75.8**	76.5
+Label	**77.3**	**79.5**	**80.0**
+Input	**79.4**	**80.8**	**80.9**

Table 4: **Average performance with different negative in-context examples.** **Red** results indicate that our method brings improvement over the regular decoding, while **Green** results denote no improvement.

	SST2	CR	SST5	Subj	QNLI	MNLI	AGNEWS
KL_divergence	0.64	0.48	0.43	0.49	0.04	0.27	0.79

Table 5: **The average KL divergence** between the normalized output distributions with positive and negative in-context examples with Llama3.2-8B.

Effects of Different number of shots. We gradually increase the number of in-context examples (denoted as N) from 1 to 16 to verify the influence of the number of shots in our method. Figure 2 reports the average performance of 7 NLU tasks and the different task QNLI. We see that our method can consistently outperform the regular decoding method with a different number of shots on aver-

Dataset	Method	α				
		0.0	0.5	1.0	1.5	2.0
SST5	**Random**	48.0	49.1	49.3	49.3	**49.5**
	BM25	53.0	**53.6**	53.3	53.2	53.1
	TopK	53.0	53.2	**53.2**	52.9	52.5
MNLI	**Random**	65.3	66.8	67.5	67.8	**68.1**
	BM25	65.8	66.6	67.1	67.4	**67.6**
	TopK	65.9	67.0	67.4	**67.7**	67.7

Table 6: **The SST5 and MNLI performance with different α.**

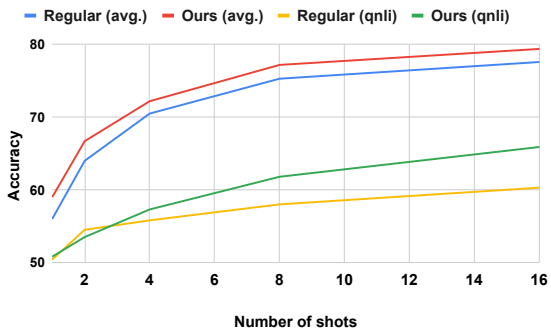


Figure 2: **The performance with different shots.**

age. For the task QNLI, as the number of shots increases, the performance gains of our method also improve. We attribute this to the model acquiring more input-label mapping information from the demonstrations, which aligns with previous findings (Pan et al., 2023).

Effects of α. The factor α in Eq. 2, which controls the importance of input-label mapping information, is an important hyper-parameter. In this part, we analyze its influence by evaluating the performance on SST5 and MNLI varying α from 0 to 2. The results on Table 6 show that: 1) the performance improves with the increase of α, and it becomes stable when $\alpha \geq 1.0$, we set $\alpha = 1$ as default; 2) For advanced demonstration selection methods(e.g.TopK), too large positive α values lead to performance degradation.

6 Conclusion

Large language models suffer from insufficient attention to the input-label mapping compared to their prior knowledge in in-context learning, leading to an unfaithful generation of the input query. In this work, we present a simple yet effective in-context contrastive decoding method that highlights input-label mapping by contrasting positive and negative in-context examples. Our experiments across various datasets and model architectures demonstrate the effectiveness and broad applicability of our approach, confirming its potential to enhance in-context learning.

Limitations

While the results presented in this paper demonstrate the effectiveness of our In-Context Contrastive Decoding (ICCD) method, there are a few limitations that warrant future exploration. First, our experiments were conducted on models up to 8B parameters, primarily due to computational limitations. Extending our method to even larger models (e.g., 70B parameters) could provide further insights into its scalability and effectiveness. Second, while our method shows promise across various Natural Language Understanding (NLU) tasks, its performance in specialized domains, such as legal or medical texts, has yet to be thoroughly examined. Third, our method requires additional forward passes to compute contrastive distributions. Although these passes are executed in parallel, the overall inference time may still increase. Future work will explore the generalizability of ICCD to these domains, as well as investigate its interaction with domain-specific datasets. Additionally, while we focused on classification tasks, other NLP tasks like text generation, machine translation, and summarization remain unexplored.

Acknowledges

We are grateful to the anonymous reviewers and the area chair for their insightful comments and suggestions. This work is supported by the National Natural Science Foundation of China (No. 62377002). This project is supported by the National Research Foundation, Singapore, under its NRF Professorship Award No. NRF-P2024-001.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child,

Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint*.

Taehyeon Kim, Joonkee Kim, Gihun Lee, and Se-Young Yun. 2024. Instructive decoding: Instruction-tuned large language models are self-refiner from noisy instructions. In *ICLR*.

Jannik Kossen, Yarin Gal, and Tom Rainforth. 2024. In-context learning learns label relationships but is not conventional learning. In *ICLR*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *ACL*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *DeeLIO*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *EMNLP*.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. 2023. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In *ACL Findings*.

Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2024. Revisiting demonstration selection strategies in in-context learning. In *ACL*.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards making the most of chatgpt for machine translation. In *Findings of EMNLP*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*.

Rico Sennrich, Jannis Vamvas, and Alireza Mohammadshahi. 2024. Mitigating hallucinations and off-target machine translation with source-contrastive and language-contrastive decoding. In *EACL*.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *NAACL*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013a. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *EMNLP*.

Shuai Wang, Liang Ding, Li Shen, Yong Luo, Zheng He, Wei Yu, and Dacheng Tao. 2024. Uscd: Improving code generation of llms by uncertainty-aware selective contrastive decoding. *arXiv preprint*.

Shuai Wang, Liang Ding, Yibing Zhan, Yong Luo, Zheng He, and Dapeng Tao. 2025. Leveraging metamemory mechanisms for enhanced data-free code generation in llms. *arXiv preprint*.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint*.

Kevin Christian Wibisono and Yixin Wang. 2024. In-context learning from training on unstructured data: The role of co-occurrence, positional information, and training data structure. In *ICML 2024 Workshop on Theoretical Foundations of Foundation Models*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing

Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report.

Zheng Zhao, Emilio Monti, Jens Lehmann, and Haytham Assem. 2024. Enhancing contextual understanding in large language models through contrastive decoding. In *NAACL*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *ICML*.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2024. ROSE doesn't do that: Boosting the safety of instruction-tuned large language models with reverse prompt contrastive decoding. In *Findings of ACL*.

A Datasets

Natural Language Understanding (NLU) Dataset information is detailed in Table 7. All NLU datasets are loaded from the HuggingFace Hub. For most NLU datasets, we report the results on the test set; while for the datasets MNLI and QNLI, we report the results on the validation set due to restricted access to their test sets.

B Templates

The templates of NLU tasks used in this paper are detailed in Table 8.

Dataset	Task	# of Classes	Data Split
SST-2	Sentiment Classification	2	6920/872/1821
SST-5	Sentiment Classification	5	8544/1101/2210
CR	Sentiment Classification	2	3394/0/376
Subj	Subjectivity Analysis	2	8000/0/2000
AgNews	Topic Classification	4	120000/0/7600
MNLI	Natural Language Inference	3	392702/19647/19643
QNLI	Natural Language Inference	2	104743/5463/5463

Table 7: **Details of NLU datasets.**

Task	Prompt	Class
SST-2	Review: "<X>" Sentiment: positive	positive
	Review: "<X>" Sentiment: negative	negative
SST-5	Review: "<X>" Sentiment: terrible	terrible
	Review: "<X>" Sentiment: bad	bad
	Review: "<X>" Sentiment: okay	okay
	Review: "<X>" Sentiment: good	good
	Review: "<X>" Sentiment: great	great
Subj	Input: "<X>" Type: objective	objective
	Input: "<X>" Type: subjective	subjective
CR	Review: "<X>" Sentiment: positive	positive
	Review: "<X>" Sentiment: negative	negative
AgNews	Input: "<X>" Type: world	World
	Input: "<X>" Type: sports	Sports
	Input: "<X>" Type: business	Business
	Input: "<X>" Type: technology	Sci/Tech
MNLI	Premise: <C> Hypothesis: <X> Prediction: entailment	Entailment
	Premise: <C> Hypothesis: <X> Prediction: neutral	Neutral
	Premise: <C> Hypothesis: <X>? Prediction: contradiction	Contradiction
QNLI	<C> Can we know <X>? Yes.	Entailment
	<C> Can we know <X>? No.	Contradiction

Table 8: **Templates of NLU tasks.** Placeholders (e.g., <X> and <C>) will be replaced by real inputs.

Different Speech Translation Models Encode and Translate Speaker Gender Differently

Dennis Fucci[α,β], **Marco Gaido**[β], **Matteo Negri**[β],
Luisa Bentivogli[β], **André F. T. Martins**[γ,δ,ϵ], **Giuseppe Attanasio**[ϵ]
[α]University of Trento, Italy [β]Fondazione Bruno Kessler, Italy
[γ]Unbabel, Portugal [δ]Instituto Superior Técnico, Portugal
[ϵ]Instituto de Telecomunicações, Portugal
dfucci@fbk.eu

Abstract

Recent studies on interpreting the hidden states of speech models have shown their ability to capture speaker-specific features, including gender. Does this finding also hold for speech translation (ST) models? If so, what are the implications for the speaker's gender assignment in translation? We address these questions from an interpretability perspective, using probing methods to assess gender encoding across diverse ST models. Results on three language directions (English → French/Italian/Spanish) indicate that while traditional encoder-decoder models capture gender information, newer architectures—integrating a speech encoder with a machine translation system via adapters—do not. We also demonstrate that low gender encoding capabilities result in systems' tendency toward a masculine default, a translation bias that is more pronounced in newer architectures.[1]

1 Introduction

Recent research in speech representation learning shows that models capture phonetic and speaker-related features in their internal representations (e.g., Prasad and Jyothi, 2020; Cormac English et al., 2022; Pasad et al., 2023; Yang et al., 2023; Choi et al., 2024; Shen et al., 2024; Chowdhury et al., 2024; Waheed et al., 2024). Building on sociophonetic studies that show how sociocultural and physiological factors shape voice production and perception (e.g., Coleman, 1976; Fuchs and Toda, 2010; Azul, 2013; Leung et al., 2018), recent research has explored how speaker *gender* is encoded in speech models. Studies have found evidence of gender encoding in self-supervised models (Chowdhury et al., 2024; de Seyssel et al., 2022; Guillaume et al., 2024) as well as in automatic speech recognition systems (Krishnan et al., 2024; Attanasio et al., 2024). Despite growing interest, an analysis of gender encoding and its impact in speech translation (ST) remains largely absent.

Gender plays a crucial role in ST, particularly when translating from *notional* to *grammatical* gender languages. Here, models must infer gender from context and correctly apply inflections. Research suggests that encoder-decoder ST systems may use acoustic cues to assign grammatical gender to words referring to the speaker (e.g., En: "I was born in..." → Fr: "Je suis né/née à..."), yet masculine defaults remain common (Bentivogli et al., 2020; Gaido et al., 2020).[2] Moreover, as the ST field evolves, traditional encoder-decoder systems are being supplemented by newer architectures that integrate pretrained speech encoders into machine translation models via adapters (speech+MT). This raises key questions: ***whether* gender information is encoded and used by current ST systems**, and ***how* architectural variations influence this process**. Addressing these questions can shed light on gender-biased behaviors, such as the systematic preference for masculine defaults.

We investigate these aspects using probing, an established interpretability method (Conneau et al., 2018; Belinkov, 2022). We train probes to predict the speaker's gender[3] from hidden states in traditional encoder-decoder and speech+MT models. Then, by analyzing translations from English into Spanish, French, and Italian, we assess how gender encoding influences the translation of gender-marked terms referring to the speaker. Our results demonstrate that encoder-decoder models encode gender information, whereas speech+MT models encode it minimally, or do not encode it at all.

[1]Code available under Apache 2.0 at https://github.com/hlt-mt/speech-translation-gender .

[2]While inferring gender from acoustic cues is common—even among humans—it should not be treated as the default. This paper examines whether models exhibit similar behavior (for further discussion, see §8).

[3]We limit our analyses to binary gender, though we recognize that gender exists on a continuum (see §8).

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1005–1019
July 27 - August 1, 2025

Moreover, the ability to extract gender from hidden states correlates with how accurately models inflect gender in speaker-referred words. Strong probing performance is a proxy for high gender translation accuracy, while weak probing performance correlates with translations defaulting to masculine. These findings suggest that ST models do not uniformly rely on acoustic cues to translate speaker-referred expressions.

Our findings are compelling for several reasons. On the one hand, they show that encoding gender yields fairer systems for female users, challenging the conventional understanding in NLP that scrubbing gender information leads to fairer outcomes (Sun et al., 2019). On the other hand, relying on biometric markers for automatic decision-making contrasts broader and established ethical principles. We discuss the matter in §8.

2 Gender Probing

The encoder of a speech model maps an arbitrary audio signal into a sequence of d-dimensional hidden states $\mathbf{X} = \langle \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L \rangle$, with $\mathbf{x}_l \in \mathbb{R}^d$ being the model's speech representation at position l and L the output sequence length. To evaluate whether and to which extent $\mathbf{X}$ encodes a specific attribute, classification *probes* (Belinkov, 2022) can be trained on states in $\mathbf{X}$, with classification accuracy indicating how well the attribute is encoded.

To obtain a single classification label for the whole sequence, previous works on gender encoding reduce its length L using mean or max pooling before training logistic probes (de Seyssel et al., 2022; Chowdhury et al., 2024; Guillaume et al., 2024; Krishnan et al., 2024) or non-linear probes (Krishnan et al., 2024). However, pooling may obscure positional variations in gender encoding, potentially weakening classification performance. Alternative approaches train separate probes on individual states $\mathbf{x}_l$ at specific positions l, assuming gender is not uniformly encoded across L. While more fine-grained than pooling, these methods have only considered a fixed, small set of relative positions (Krishnan et al., 2024) or do not support sequences of arbitrary length (Attanasio et al., 2024).

To avoid the limitations above, we design an attention-based probe. We draw inspiration from the Q-Former (Li et al., 2023), which maps a sequence of variable length into a fixed number of vectors by means of an attention mechanism where the query is a fixed sequence of learnable vectors. Similarly, our probe leverages a single learnable query to selectively extract gender information from hidden states. The sequence of hidden states $\mathbf{X}$ is projected into key ($\mathbf{K} \in \mathbb{R}^{L \times d}$) and value ($\mathbf{V} \in \mathbb{R}^{L \times d}$) matrices using learnable weight matrices $\mathbf{W}_K \in \mathbb{R}^{d \times d}$ and $\mathbf{W}_V \in \mathbb{R}^{d \times d}$, respectively. We then compute scaled-dot attention using $\mathbf{K}$, $\mathbf{V}$, and a learnable query vector $\mathbf{q} \in \mathbb{R}^d$. The output representation $\mathbf{o} \in \mathbb{R}^d$ is passed through a linear layer to compute class logits, and final probabilities are obtained via softmax.[4] The attention weights $\mathbf{a} \in \mathbb{R}^L$ instead indicate, for each input sequence, which hidden states contribute most to gender encoding. By using attention to selectively extract gender information from the entire input, followed by a single linear classification layer, the architecture remains simple—an important property for avoiding spurious correlations, as recommended for probing classifiers (Belinkov, 2022). At the same time, this attention-based design is more expressive than basic linear models (Hewitt and Liang, 2019), and serves as a proxy for how ST decoders access encoder states—namely, through the cross-attention mechanism. For a comparison with probes from previous works, see Appendix A.

3 Experimental Setup

Models We evaluate one ST encoder-decoder and two speech+MT models (full details in Appendix B) for en→es/fr/it translation. As encoder-decoder model (`enc-dec`), we use a Transformer-based model by Wang et al. (2020), trained on ST data from MuST-C (Cattoni et al., 2021). The speech+MT models are SeamlessM4T (`Seamless`—Communication et al., 2023) and ZeroSwot (`ZeroSwot`—Tsiamas et al., 2024). We probe the model states at different locations. In `enc-dec`, we use the encoder outputs. In speech+MT models, to assess the impact of adapters—which compress and map the speech encoder's outputs to the MT model's embedding space—we probe hidden states before (`pre-ad`) and after (`post-ad`) the adapters.

Data To train and evaluate probes, we use the MuST-C corpus, which includes speaker gender annotations based on self-declared pronouns (`He/She`) at the time of data collection (Gaido et al., 2020), thereby avoiding the risk of misgendering (see §8). For training, we randomly sample audio segments from the MuST-C training set (en→es section) to create gender-balanced training (`train`)

[4] See Appendix B for implementation details.

		test-generic	test-speaker									
			en→es			en→fr			en→it			Avg.
		All	She	He	All	She	He	All	She	He	All	All
Seamless	post-ad	59.76	90.41	24.30	51.72	86.18	29.55	54.51	85.98	32.14	55.63	53.95
	pre-ad	75.53	85.61	51.41	67.32	81.25	54.64	67.52	83.76	58.57	70.58	68.47
ZeroSwot	post-ad	59.57	85.98	38.73	61.80	90.13	37.80	61.62	86.35	39.64	60.65	61.36
	pre-ad	89.60	96.68	84.15	90.25	95.72	84.19	90.02	92.99	84.29	88.56	89.61
enc-dec		92.21	99.26	87.32	93.14	99.67	89.35	94.59	98.89	93.57	96.19	94.64

Table 1: Scores for gender probing (macro F1 for All, recall for She/He) across all configurations.

and validation (dev) sets. For testing, we use two datasets.[5] Test-generic consists of generic utterances taken from the validation and test sets of MuST-C. Since these sets are imbalanced across gender classes, test-generic is also inherently unbalanced. Test-speaker is drawn from MuST-SHE (Bentivogli et al., 2020), a corpus specifically designed to evaluate gender translation related to human referents from English into Spanish, French, and Italian. In MuST-SHE reference translations, each gender-marked word—corresponding to a neutral expression in English—is annotated with its opposite (wrong) gender form. We use the portion of MuST-SHE containing first-person references (e.g. En. "I was born" → Fr. "Je suis née^{F}<né^{M}>"), which is balanced across the two genders. Through test-speaker, we test gender encoding, gender translation, and their relationship, as the only cue for determining gender translation is the speaker's voice characteristics.

Evaluation We use macro F1 for overall probing performance, and recall for individual balanced classes. For overall ST quality, we use COMET (Rei et al., 2020).[6] For gender translation, we use the MuST-SHE evaluation script. It calculates *accuracy* as the percentage of speaker-referred words generated with the correct gender, out of all generated speaker-referred words. Instances where the model fails to predict the speaker-referred words—regardless of gender—are excluded from the accuracy calculation. The percentage of words included in the accuracy computation is reported as *coverage* (Gaido et al., 2020). Although out-of-coverage instances are excluded from gender accuracy computations, they can still convey gender information—for example, when the generated term is a gendered synonym. However, through manual analysis of MuST-SHE translations by ST systems, Savoldi et al. (2022) demonstrated that out-of-coverage cases are typically either gendered synonyms that follow the same accuracy patterns as in-coverage examples, or mistranslations, where gender assessment is not applicable. We confirmed that this pattern also holds for the ST models investigated in this study through a manual analysis of out-of-coverage outputs, detailed in Appendix C.

4 Results

4.1 Gender Probing

Table 1 presents the macro F1 and single-class recall scores for the probes on test-generic and test-speaker. We first compare the results of probes tested on the final speech representations of enc-dec and Seamless/ZeroSwot post-ad. Overall, probes trained on the hidden states of enc-dec achieve high accuracy, with F1 scores peaking at 96.19 on test-speaker (All set, en→it) and scoring 92.21 on test-generic. In contrast, probes trained on Seamless/ZeroSwot post-ad yield significantly lower scores, not exceeding 59.76 on test-generic and 61.80 on test-speaker (en→es), respectively. Focusing on test-speaker, we note that ZeroSwot post-ad retains slightly more gender information than Seamless post-ad. Single-class recall scores are higher for She than He across all probes, especially in the weakest probe trained on Seamless post-ad. While this might suggest that the He class is harder to extract, the strong skew toward the She class in Seamless and ZeroSwot post-ad likely results from the probes' overall difficulty in extracting gender from these hidden states. Overall, these findings indicate that **speaker's gender encoding capability is high in the speech representations of the traditional encoder-decoder model, but significantly lower in speech+MT models**.

We now analyze the impact of the adapters in

[5] No speaker overlaps exist across train, dev, and test sets. For details on data statistics, see Appendix D.

[6] We use the Unbabel/wmt22-comet-da model.

		She			He			All		
		COMET	Cov.	Acc.	COMET	Cov.	Acc.	COMET	Cov.	Acc.
Seamless	**en→es**	80.81	70.37	12.09	81.69	70.56	90.55	81.26	70.47	53.62
	en→fr	78.31	59.61	17.60	79.50	59.66	88.84	78.89	59.63	53.29
	en→it	79.35	57.11	13.30	82.45	57.52	91.36	80.93	57.32	53.15
	avg.	79.49	62.36	14.33	81.21	42.61	90.25	80.36	62.47	53.35
ZeroSwot	**en→es**	85.57	78.04	51.50	83.56	73.72	76.73	84.54	75.79	64.46
	en→fr	82.99	65.52	54.38	82.20	64.55	70.11	82.60	65.03	62.20
	en→it	84.62	65.23	46.18	84.71	60.19	77.87	84.67	62.66	61.75
	avg.	84.39	69.60	50.69	83.49	66.15	74.90	83.94	67.83	62.80
enc-dec	**en→es**	77.03	66.93	81.47	75.64	65.69	90.94	76.32	66.29	86.45
	en→fr	71.77	53.20	77.68	73.68	53.79	91.34	72.70	53.50	84.62
	en→it	74.56	55.33	76.44	76.02	54.85	94.47	75.30	55.09	85.65
	avg.	74.45	58.49	78.53	75.11	58.11	92.25	74.77	58.29	85.57

Table 2: Translation performance in quality (COMET), gender coverage, and accuracy on `test-speaker` for all three language pairs, along with average scores.

`Seamless` and `ZeroSwot`. Looking at `pre-ad` results, `ZeroSwot` exhibits stronger probing performance compared to `Seamless` (avg. 89.61 vs 68.47 on `test-speaker`). This difference may stem from their training strategies: `Seamless` is jointly trained on speech and text, while `ZeroSwot` trains only the speech encoder, keeping the rest of the model frozen. Comparing `pre-ad` and `post-ad` results, we observe a substantial decrease in probing performance, with a $\sim 21\%$ drop in F1 for `Seamless` and $\sim 32\%$ for `ZeroSwot` after the adapters, across both test sets. These results indicate that **mapping speech representation into the MT embedding space via adapters significantly removes gender information in the speech+MT models.**

In summary, gender encoding varies across models. Speech+MT systems show lower encoding capability, particularly after the adapters, while the encoder-decoder model shows higher encoding capability, similarly to speech models explored in previous works. Additionally, we conducted an analysis to examine how gender information is distributed across the sequence length (see Appendix E). Interestingly, and consistent with prior findings on ASR models, we observe that ST models primarily encode gender in the early positions of the sequence.

4.2 Speaker's Gender Translation

Table 2 presents translation scores on `test-speaker` across the three ST models and language directions. Overall, `ZeroSwot` achieves the best translation quality, followed by `Seamless` and `enc-dec` (avg. COMET scores are respectively 83.94, 80.36, and 74.77). Coverage scores align with COMET trends, indicating that higher translation quality increases the likelihood that system outputs include the speaker-referred words annotated in the reference. In all cases, more than half of the annotated words can be evaluated for gender translation.

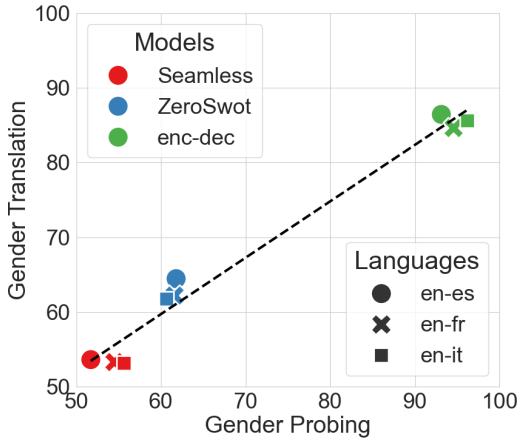


Figure 1: Correlation between overall gender probing performance (macro F1) and gender translation accuracy across models and languages on `test-speaker`.

Regarding gender translation accuracy, `enc-dec` provides the highest scores (avg. 85.57), while `ZeroSwot` and `Seamless` register lower average scores of 62.80 and 53.35, respectively. Notably, speech+MT models with stronger overall translation quality exhibit lower accuracy. Instead, as shown in Figure 1, accuracy scores strongly correlate with probing performance ($R^2 = 0.99$, $p < 0.01$). In other words, **higher gender encoding capability leads to more accurate gender translation**, supporting the idea that ST models rely on acoustic cues to translate speaker-referred

words when other signals are absent. When gender information is minimal, translations skew masculine. For instance, `Seamless`, for which gender encoding is almost absent (low F1 scores in Table 1), strongly favors masculine forms, with feminine score only peaking at 17.60 (en→fr). A similar bias is observed in `ZeroSwot`, where average accuracy is 50.69 for the `She` class and 74.90 for the `He` class. Even `enc-dec` exhibits a masculine skew: although it achieves higher feminine accuracy compared to the other models (avg. 78.53), this remains significantly lower than its masculine score (avg. 87.62), despite clear evidence of gender encoding (see Table 1). This suggests that an underlying linguistic bias sometimes overrides acoustic cues (see Appendix F for some examples).

All in all, ST models show variable performance on gender translation, with gender encoding serving as a proxy for the agreement between translation and the speaker's gender. When gender encoding is low, a masculine bias emerges.

5 Conclusion

We investigated how diverse ST architectures encode speaker gender information. Using attention-based probes, we found that while traditional encoder-decoder models trained solely on ST data retain this information, the adapters in newer speech+MT architectures tend to erase it. Moreover, gender encoding is correlated with the ST model's ability to assign the correct grammatical gender to words referring to the speaker. This finding suggests that ST models can leverage acoustic gender information when available. When such information is weakly encoded, the models default to masculine translations more frequently.

Our study sheds new light on how ST models encode and use gender information during translation, opening avenues for further research. For example, future work could explore which properties of adapters—or aspects of their training—contribute to the loss of gender information, and whether similar patterns emerge for acoustic and paralinguistic features that may be weakened during the mapping to the text embedding space. These findings also inform strategies to mitigate biases such as undue masculine translations. One approach could involve modifying the configuration of the adapters to better preserve the notion of the speaker's gender—when appropriate—or integrating external knowledge to avoid misgendering.

6 Acknowledgments

The work of Marco Gaido was funded by the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU. The work of Luisa Bentivogli and Matteo Negri was supported by the European Union's Horizon research and innovation programme under grant agreement No 101135798, project Meetween (My Personal AI Mediator for Virtual MEETtings BetWEEN People). The work of André F.T. Martins and Giuseppe Attanasio was supported by EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), by the project DECOLLAGE (ERC-2022-CoG 101088763), by the Portuguese Recovery and Resilience Plan through project C645008882-00000055 (Center for Responsible AI), and by FCT/MECI through national funds and when applicable co-funded EU funds under UID/50008: Instituto de Telecomunicações.

7 Limitations

Models and Languages Our analysis focuses on specific models and language directions. The choice of models was driven by their performance and widespread use in the selected language pairs. The selection of languages was based on the need to translate gender-neutral forms into gender-marked ones and the availability of annotated ST data. As more data becomes available, we plan to expand our analysis to include additional translation systems and languages, as well as systems which integrate speech into Large Language Models (Gaido et al., 2024). Nevertheless, prior research (Guillaume et al., 2024; Attanasio et al., 2024) suggests that gender encoding mechanisms are consistent across languages, so it is likely that similar patterns would emerge in other languages.

Probing While the probing paradigm (Conneau et al., 2018) is intuitive and well-established, it has been criticized for not directly confirming whether the model uses the extracted information (Belinkov, 2022). For example, some studies have highlighted a mismatch between probe performance and the original model's performance in NLP tasks (Belinkov and Glass, 2019; Ravichander et al., 2021; Elazar et al., 2021). Our experiments demonstrate a strong correlation between gender classification through probes (as an auxiliary task) and a specific translation aspect: gender assignment to words re-

ferring to the speaker (the original *sub*task). Although correlation does not imply causation, this finding proves that ST models likely use acoustic information for gender translation. To build on this, in future work, we plan to explore further interpretability techniques, such as *amnesic probing* (Elazar et al., 2021) or other methods in the field of mechanistic interpretability (Ferrando et al., 2024; Saphra and Wiegreffe, 2024).

8 Ethics Statement

Following prior work (de Seyssel et al., 2022; Guillaume et al., 2024; Chowdhury et al., 2024; Krishnan et al., 2024; Attanasio et al., 2024), we use the term "gender" as an approximation to account for documented differences in voices *and* linguistic expression of gender identity (She or He). Adopting this framework, we investigated how ST models encode gender-related vocal differences, which may stem from both physiological factors—such as anatomical differences in vocal tracts between male and female speakers (Simpson, 2001; Hillenbrand and Clark, 2009)—and sociocultural aspects—such as vocal behaviors associated with masculinity or femininity (Coleman, 1976; Nylén et al., 2024). Accordingly, to analyze how gender encoding capabilities affect ST systems' bias, we adopted a binary framework with only She/He class labels, primarily due to the lack of extensive speech data featuring non-binary voices. However, we recognize that vocal differences related to gender exist on a continuum, just as gender identities do. Expanding research to encompass non-binary identities is an essential next step.

In the context of translation, *gender bias* refers to the tendency of systems to favor one gender form over another—typically masculine over feminine—or to associate translations of specific roles and professions with a particular gender based on stereotypes (Savoldi et al., 2021). These biases can affect users' self-perception, as gendered language plays a fundamental role in shaping personal identity and representation (Stahlberg et al., 2007; Corbett, 2013; Gygax et al., 2019). As noted by Blodgett et al. (2020) and Savoldi et al. (2021), gender bias in translation technologies can lead to representational harms, such as reducing the visibility of women or reinforcing negative stereotypes about gender groups, as well as allocational harms, including disparities in the quality of service received by male and female users. Our experiments show that some models exhibit a strong bias toward masculine forms, posing a risk of such harms. We also find that encoding vocal characteristics related to gender can affect the accuracy of speaker gender translation.

Leveraging acoustic information was a key motivation behind the shift from cascaded systems—where vocal information was lost between ASR and MT—to direct models aimed at improving ST quality (Sperber and Paulik, 2020; Bentivogli et al., 2021). Our study provides a concrete example, demonstrating that accuracy of gender translation improves when gender-related vocal differences are encoded by the ST model. However, we do not necessarily advocate leveraging vocal features to assign linguistic gender markers, as such decisions may not align with a speaker's gender identity. This concern is particularly relevant for transgender individuals, children, and people with vocal impairments (Matar et al., 2016; Pereira et al., 2018; Villas-Bôas et al., 2021; Menezes et al., 2022). To avoid the risk of misgendering in our evaluation, we relied on self-declared gender identities. Although vocal properties in real-world scenarios may lead to misgendering and should be interpreted with caution, we argue that our findings nonetheless provide valuable insights. On one hand, our work contributes to advancing the understanding of gender bias in ST. On the other hand, raising awareness of how ST models encode speaker-related information—and how this influences translations—can help stakeholders make informed decisions to prevent harmful misgendering. It also empowers developers to make deliberate choices regarding model architecture and training strategies, as we found that specific architectures affect how gender information is encoded and used. Therefore, by enhancing the interpretability of ST models, we believe our work contributes to positive social impact.

References

Giuseppe Attanasio, Beatrice Savoldi, Dennis Fucci, and Dirk Hovy. 2024. Twists, humps, and pebbles: Multilingual speech recognition models exhibit gender performance gaps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21318–21340, Miami, Florida, USA. Association for Computational Linguistics.

David Azul. 2013. How do voices become gendered? a critical examination of everyday and medical con-

structions of the relationship between voice, sex, and gender identity. In *Challenging popular myths of sex, gender and biology*, pages 77–88. Springer.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Preprint*, arXiv:2006.11477.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Luisa Bentivogli, Mauro Cettolo, Marco Gaido, Alina Karakanta, Alberto Martinelli, Matteo Negri, and Marco Turchi. 2021. Cascade versus Direct Speech Translation: Do the Differences Still Make a Difference? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2873–2887, Online. Association for Computational Linguistics.

Luisa Bentivogli, Beatrice Savoldi, Matteo Negri, Mattia A. Di Gangi, Roldano Cattoni, and Marco Turchi. 2020. Gender in danger? evaluating speech translation technology on the MuST-SHE corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6923–6933, Online. Association for Computational Linguistics.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. MuST-C: A multilingual corpus for end-to-end speech translation. *Computer Speech & Language*, 66:101155.

Kwanghee Choi, Ankita Pasad, Tomohiko Nakamura, Satoru Fukayama, Karen Livescu, and Shinji Watanabe. 2024. Self-supervised speech representations are more phonetic than semantic. In *Interspeech 2024*, pages 4578–4582.

Shammur Absar Chowdhury, Nadir Durrani, and Ahmed Ali. 2024. What do end-to-end speech models learn about speaker, language and channel information? a layer-wise and neuron-level analysis. *Computer Speech & Language*, 83:101539.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. *Preprint*, arXiv:2108.06209.

Ralph O. Coleman. 1976. A comparison of the contributions of two voice quality characteristics to the perception of maleness and femaleness in the voice. *Journal of Speech & Hearing Research*, 19(1):168–180.

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, et al. 2023. SeamlessM4T: Massively Multilingual & Multimodal Machine Translation. *Preprint*, arXiv:2308.11596.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Greville G. Corbett. 2013. *The Expression of Gender*. De Gruyter.

Patrick Cormac English, John D. Kelleher, and Julie Carson-Berndsen. 2022. Domain-informed probing of wav2vec 2.0 embeddings for phonetic features. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 83–91, Seattle, Washington. Association for Computational Linguistics.

Maureen de Seyssel, Marvin Lavechin, Yossi Adi, Emmanuel Dupoux, and Guillaume Wisniewski. 2022. Probing phoneme, language and speaker information in unsupervised speech representations. In *Interspeech 2022*, pages 1402–1406.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. 2024. A Primer on the Inner Workings of Transformer-based Language Models. *Preprint*, arXiv:2405.00208.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. 2015. Learning with a wasserstein loss. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2053–2061, Cambridge, MA, USA. MIT Press.

Susanne Fuchs and Martine Toda. 2010. Do differences in male versus female/s/reflect biological or sociophonetic factors. *Turbulent sounds: An interdisciplinary guide*, 21:281–302.

Marco Gaido, Sara Papi, Matteo Negri, and Luisa Bentivogli. 2024. Speech Translation with Speech Foundation Models and Large Language Models: What is There and What is Missing? *Preprint*, arXiv:2402.12025.

Marco Gaido, Beatrice Savoldi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2020. Breeding gender-aware direct speech translation systems. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3951–3964, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Séverine Guillaume, Maxime Fily, Alexis Michaud, and Guillaume Wisniewski. 2024. Gender and language identification in multilingual models of speech: Exploring the genericity and robustness of speech representations. In *Interspeech 2024*, pages 3330–3334.

Pascal M. Gygax, Daniel Elmiger, Sandrine Zufferey, Alan Garnham, Sabine Sczesny, Friederike von Stockhausen, Lisa Braun, and Jane Oakhill. 2019. A language index of grammatical gender dimensions to study the impact of grammatical gender on the way we perceive women and men. *Frontiers in Psychology*, 10.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

James M. Hillenbrand and Michael J. Clark. 2009. The role of f0 and formant frequencies in distinguishing the voices of men and women. *Attention Perception & Psychophysics*, 71(5):1150–1166.

Aravind Krishnan, Badr M. Abdullah, and Dietrich Klakow. 2024. On the encoding of gender in transformer-based asr representations. In *Interspeech 2024*, pages 3090–3094.

Yeptain Leung, Jennifer Oates, and Siew Pang Chan. 2018. Voice, articulation, and prosody contribute to listener perceptions of speaker gender: A systematic review and meta-analysis. *Journal of Speech, Language, and Hearing Research*, 61(2):266–297.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Nayla Matar, Cristel Portes, Leonardo Lancia, Thierry Legou, and Fabienne Baider. 2016. Voice quality and gender stereotypes: A study on Lebanese women with Reinke's edema. *Journal of Speech, Language, and Hearing Research*, 59(6):1608–1617.

Danielle Pereira Menezes, Zulina Souza de Lira, Ana Nery Barbosa de Araújo, Anna Alice Figueirêdo de Almeida, Adriana de Oliveira Camargo Gomes, Bruno Teixeira Moraes, and Jonia Alves Lucena. 2022. Prosodic differences in the voices of transgender and cisgender women: Self-perception of voice - an auditory and acoustic analysis. *Journal of Voice*.

Frida Nylén, Johan Holmberg, and Maria Södersten. 2024. Acoustic cues to femininity and masculinity in spontaneous speech. *The Journal of the Acoustical Society of America*, 155(5):3090–3100.

Ankita Pasad, Bowen Shi, and Karen Livescu. 2023. Comparative layer-wise analysis of self-supervised speech models. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: an imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(null):2825–2830.

Amanda Maria Pereira, Ana Paula Dassie-Leite, Eliane Cristina Pereira, Juliana Benthien Cavichiolo, Marcelo de Oliveira Rosa, and Elmar Allen Fugmann. 2018. Auditory perception of lay judges about gender identification of women with reinke's edema. *CoDAS*, 30(4).

Gabriel Peyré and Marco Cuturi. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Archiki Prasad and Preethi Jyothi. 2020. How accents confound: Probing for accent information in end-to-end speech recognition systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3739–3753, Online. Association for Computational Linguistics.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Naomi Saphra and Sarah Wiegreffe. 2024. Mechanistic? In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 480–498, Miami, Florida, US. Association for Computational Linguistics.

Beatrice Savoldi, Marco Gaido, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. Gender bias in machine translation. *Transactions of the Association for Computational Linguistics*, 9:845–874.

Beatrice Savoldi, Marco Gaido, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2022. Under the morphosyntactic lens: A multifaceted evaluation of gender bias in speech translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1807–1824, Dublin, Ireland. Association for Computational Linguistics.

Gaofei Shen, Michaela Watkins, Afra Alishahi, Arianna Bisazza, and Grzegorz Chrupała. 2024. Encoding of lexical tone in self-supervised models of spoken language. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4250–4261, Mexico City, Mexico. Association for Computational Linguistics.

Adrian P. Simpson. 2001. Dynamic consequences of differences in male and female vocal tract dimensions. *The Journal of the Acoustical Society of America*, 109(5 Pt 1):2153–2164.

Matthias Sperber and Matthias Paulik. 2020. Speech translation and the end-to-end promise: Taking stock of where we are. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7409–7421, Online. Association for Computational Linguistics.

Dagmar Stahlberg, Friederike Braun, Lisa Irmen, and Sabine Sczesny. 2007. Representation of the sexes in language. In Klaus Fiedler, editor, *Social Communication*, pages 163–187. Psychology Press.

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1640, Florence, Italy. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, et al. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Ioannis Tsiamas, Gerard Gállego, José Fonollosa, and Marta Costa-jussà. 2024. Pushing the limits of zero-shot end-to-end speech translation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14245–14267, Bangkok, Thailand. Association for Computational Linguistics.

Anna Paula Villas-Bôas, Karine Schwarz, Anna Martha Vaitses Fontanari, Angelo Brandelli Costa, Dhiordan Cardoso da Silva, Maiko Abel Schneider, Carla Aparecida Cielo, Poli Mara Spritzer, and Maria Inês Rodrigues Lobato. 2021. Acoustic measures of brazilian transgender women's voices: A case–control study. *Frontiers in Psychology*, 12.

Abdul Waheed, Hanin Atwany, Bhiksha Raj, and Rita Singh. 2024. What do speech foundation models not learn about speech? *Preprint*, arXiv:2410.12948.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020. fairseq s2t: Fast speech-to-text modeling with fairseq. In *Proceedings of the 2020 Conference of the Asian Chapter of the Association for Computational Linguistics (AACL): System Demonstrations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Mu Yang, Ram C. M. C. Shekar, Okim Kang, and John H. L. Hansen. 2023. What can an accent identifier learn? probing phonetic and prosodic information in a wav2vec2-based accent identification model. In *Interspeech 2023*, pages 1923–1927.

Jinming Zhao, Hao Yang, Gholamreza Haffari, and Ehsan Shareghi. 2022. M-adapter: Modality adaptation for end-to-end speech-to-text translation. In *Interspeech 2022*, pages 111–115.

A Probe Comparison

In our experiments, we use an attention-based probe, introduced in §2. It is designed to predict a

	Seamless pre-ad	ZeroSwot pre-ad	Enc-Dec	avg.
max pooling	61.37	54.46	65.48	60.44
mean pooling	62.17	76.44	90.10	76.24
positional sampling	71.45	64.96	83.37	73.26
attention-based	**75.53**	**89.60**	**92.21**	**85.78**

Table 3: F1 scores of probes on test-generic across different ST model states and probing strategies.

single label for each input sequence while preserving gender encoding across the temporal dimension. Unlike traditional approaches, our method avoids sampling or pooling mechanisms, which can undermine classification performance.

To assess its effectiveness, we compare our attention-based probe with three probing methods from prior work: logistic classifiers trained on hidden representations aggregated *i)* via max pooling, *ii)* via mean pooling, and *iii)* positional sampling of hidden states at relative positions. In the latter case, we sample hidden states at every 25% of the sequence length and train five separate probes, each corresponding to one of these positions. We train these probes on the hidden states extracted from Seamless/ZeroSwot pre-ad and enc-dec, as these representations encode the speaker's gender to a meaningful—although variable—extent. Table 3 reports the corresponding F1 scores on test-generic. For positional sampling, we report the highest scores among the five positions, which consistently correspond to the first position. Interestingly, this aligns with our analysis in Appendix E, where we show that attention weight distributions tend to concentrate at early positions.

Our results show that the attention-based probe consistently outperforms all other probing strategies across all considered settings. Notably, max pooling performs the worst (avg. 60.44), while mean pooling is the closest to the attention-based solution (avg. 76.24 vs. 85.78, respectively). The positional sampling method falls between mean pooling and attention-based, even though it surpasses mean pooling on Seamless pre-ad.

Overall, our approach proves to be more effective than traditional methods in our scenario. Moreover, our probe offers the additional advantage of interpretability by providing attention weights that reveal how gender information is distributed across the sequence (see Appendix E).

B Experimental Details

ST Models Below, we describe the architectures of the models used in our study (see §3), along with the details for running inference operations and training the probes.

- Transformer-based encoder-decoder model (Wang et al., 2020): it employs a convolutional downsampler to reduce the length of speech inputs, followed by a Transformer encoder—initialized from an ASR model—and a decoder. The encoder output is used for the enc-dec setting. Both the encoder and decoder are jointly trained for the ST task using autoregressive cross-entropy loss.

- SeamlessM4T (Communication et al., 2023): it uses a speech encoder based on the pre-trained w2v-BERT 2.0 (Chung et al., 2021), whose output is used in the Seamless pre-ad setting. A length adapter, derived from the M-adaptor (Zhao et al., 2022), processes the speech encoder's hidden states, and the resulting output is taken for the Seamless post-ad setting. The speech encoder followed by the length adapter is paired with a text encoder based on the pre-trained No Language Left Behind (NLLB) encoder (Team et al., 2022). Both encoders share a common text decoder, which is initialized from the NLLB decoder. The entire model is jointly trained using cross-entropy loss for speech-to-text and text-to-text translation tasks, along with token-level knowledge distillation from MT task (teacher) to the ST task (student).

- ZeroSwot (Tsiamas et al., 2024): it encodes speech using Wav2Vec 2.0 (Baevski et al., 2020), with its representations used in the ZeroSwot pre-ad setting. A CTC module then predicts characters from these representations. The CTC probabilities and Wav2Vec features are processed through an adapter to produce a compressed acoustic representation.

This representation is then enhanced with positional encodings to form the final speech embedding, which is used in the `ZeroSwot post-ad` setting. The Wav2Vec encoder, CTC module, and adapter are fine-tuned by jointly minimizing the CTC loss and the Wasserstein loss (Frogner et al., 2015) using Optimal Transport (Peyré and Cuturi, 2019), to align speech-derived representations with the text-derived embeddings produced by the NLLB encoder.

Audio data are normalized to `float32` and truncated at 60s. Inference for all models is conducted using `transformers 4.47.0` (Wolf et al., 2020) on an NVIDIA A40 GPU (48GB RAM). We use publicly available configuration files and model checkpoints from the HuggingFace Hub: https://huggingface.co/facebook/seamless-m4t-v2-large for SeamlessM4T, https://huggingface.co/johntsi/ZeroSwot-Large_asr-mustc_mt-mustc_en-to-8 for ZeroSwot, https://huggingface.co/facebook/s2t-medium-mustc-multilingual-st for the Transformer-based encoder-decoder.

Probes We implemented our attention-based probes using `PyTorch 2.3.0` (Paszke et al., 2019). For training, we use a batch size of 32 and a starting learning rate of 0.0001, reduced by half if the validation loss does not decrease for 3 consecutive epochs. The weights of the classification layer are initialized using Xavier initialization (Glorot and Bengio, 2010). We use the Adam optimizer and optimize the model with cross-entropy loss. Training stops if the validation loss on the `dev` set does not improve by at least 0.00001 for 20 epochs. All training and testing were conducted on an NVIDIA Tesla K80 GPU (12GB RAM). For the probes used in Appendix A, we used logistic classifier implemented with `scikit-learn 1.4.1` (Pedregosa et al., 2011) (`SGDClassifier` class with `log loss` and default values).

C Manual Evaluation

For the evaluation of gender translation, we adopt the official metrics from MuST-SHE (Bentivogli et al., 2020)—*gender accuracy* and *coverage* (see §3). Coverage is the percentage of sentences for which accuracy can be assessed. This number relies on automatic string matching, i.e., by checking whether the term annotated for gender appears, in any gendered form, in the system output. Accuracy is then measured only on the *in-coverage* (IC) outputs by verifying the gender marking of the word found. Therefore, coverage is complementary to accuracy. However, IC accuracy offers only a partial view, as *out-of-coverage* (OOC) instances—excluded from gender accuracy calculations—can still convey gender information, for example, when the generated term is a gendered synonym. To assess whether OOC instances exhibit trends that align with or diverge from the accuracy scores observed in IC cases, one of the authors manually analyzed the OOC outputs for Italian translations produced by the three ST models.

The manual analysis reveals that the OOC cases include: *(i)* translations that avoid speaker-referred gendered words by using paraphrases—which can sometimes be incorrect—or epicene terms; *(ii)* gendered synonyms; and *(iii)* gendered mistranslations referring to the speaker. Gender accuracy evaluation is possible only for cases of type *(ii)* and *(iii)*. Examples of all three categories are provided below.

(i)
En. source: *And to make sure* [...]
It. reference: *Per essere* <u>*sicuro*</u>^{M} *che* [...]
`Seamless` output: *E per* <u>*assicurarmi*</u> *che* [...]
Note: The model uses the verb "assicurarmi" (En. "to make sure") instead of the expression "essere sicuro" (En. "to be sure") found in the reference.

(ii)
En. source: [...] *I was amazed* [...]
It. reference: [...] *Ero* <u>*stupefatto*</u>^{M}. [...]
`ZeroSwot` output: [...] *Ero* <u>*stupito*</u>^{M}. [...]
Note: The adjective "stupito" is a synonym of "stupefatto", both meaning "amazed".

(iii)
En. source: *I'm exhausted, and I'm numb.*
It. reference: *Sono* <u>*esausta*</u>^{F}, *e sono* <u>*paralizzata*</u>^{F}.
`Seamless` output: *Sono* <u>*esausta*</u>^{F} *e sono* <u>*sordo*</u>^{M}.

	She		He	
	Cov.	Acc.	Cov.	Acc.
	String Matching			
Seamless	57.11	13.30	57.52	91.36
ZeroSwot	65.23	46.18	60.19	77.87
enc-dec	55.33	76.44	54.85	94.47
	String Matching + Manual			
Seamless	70.56	12.59	73.79	93.75
ZeroSwot	78.43	45.95	76.94	80.76
enc-dec	71.57	78.55	70.39	95.32

Table 4: Coverage and accuracy scores for the She and He classes in the en→it direction. The upper three rows report results based solely on automatic string-matching evaluation (as in Table 2), while the lower three rows also include OOC instances where gender translation was manually evaluated.

> **Note**: The adjective "sordo" (En. "deaf") is a mistranslation of "numb" in this context.

Cases of type *(ii)* and *(iii)* account for 48.40% of the OOC instances for Seamless, 43.17% for ZeroSwot, and 64.64% for enc-dec. We incorporate these cases into the IC set and recompute the scores. Table 4 reports both the scores obtained using string-matching evaluation—as in Table 2—and those combining string matching with manual assessment of OOC instances. Coverage increases in the latter due to the reduced number of unassessable cases. More importantly, accuracy scores remain largely consistent. Absolute differences between string-matching scores and those including manual evaluation range from just 0.23 (for ZeroSwot, She class) to 2.89 (for ZeroSwot, He class). This consistency confirms that gender accuracy, when extended to include manually assessed OOC instances, aligns closely with that observed for IC cases, echoing findings by Savoldi et al. (2022) in their analysis of other systems.

Overall, this manual evaluation suggests that, despite the limitations of string-matching metrics, the trends they reveal remain broadly reliable, and the minor variations in gender accuracy introduced by manual evaluation do not affect the main findings.

D Data Statistics

Table 5 summarizes the distribution of samples, hours, and speakers across the dataset splits used for training and testing our probes.

The train set consists of 5061 samples, covering 9.70 hours of audio, with a balanced gender representation across 1417 speakers. However, the number of speakers in the He class is higher than in the She class (883 vs. 534), reflecting the inherent imbalance in the MuST-C dataset (Cattoni et al., 2021). The dev set, approximately 20% the size of train, contains 1078 samples and 2.08 hours of gender-balanced audio from 337 speakers. The test-generic set includes 3702 samples spanning 6.69 hours, with a lower speaker-to-sample ratio than train and dev. Unlike these sets, test-generic is not gender-balanced, as it was created by merging the original test and validation sets of MuST-C, where male-speaker samples are significantly more numerous. Finally, the test-speaker set is the smallest, containing over 550 samples and 1.30 hours of audio, with small variations across language directions. Unlike test-generic, this set maintains gender balance.

E Gender Encoding Across Sequence Lengths

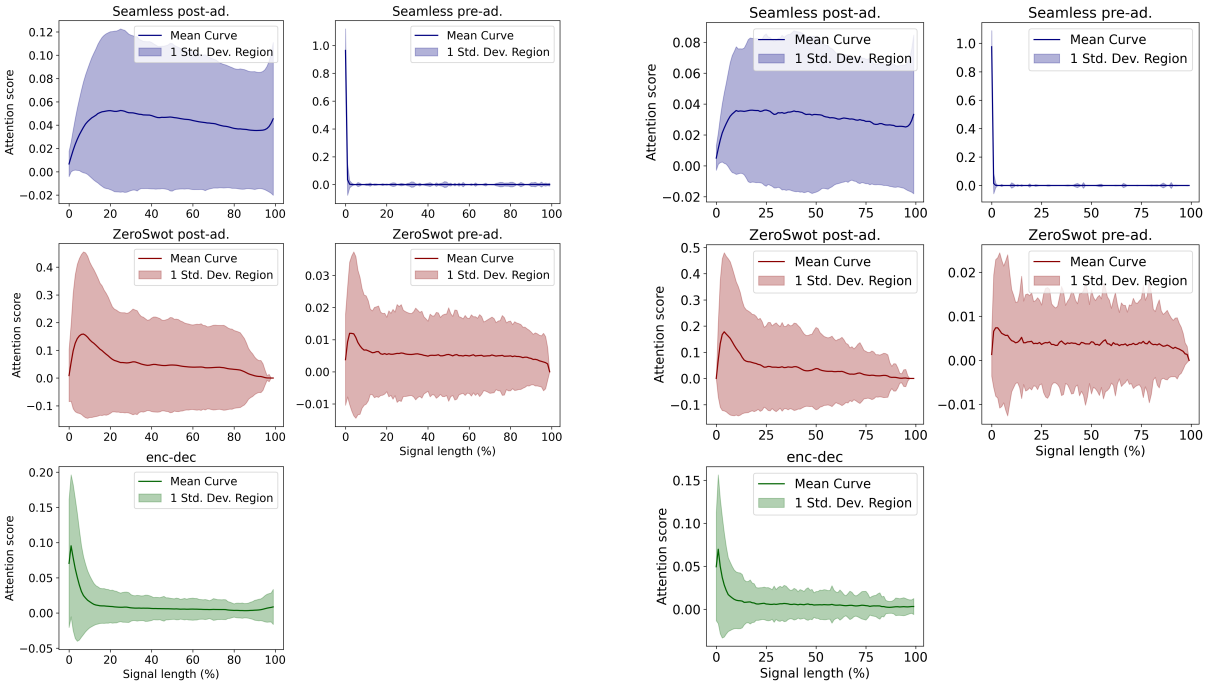
For each input sequence, our probes provide attention weights $\mathbf{a}$ that indicate the positions in the sequence where the probe primarily focuses when predicting gender (see §2). These trends can serve as a proxy for understanding how gender information is distributed across the sequence length, particularly for probes that effectively extract gender-related features. To analyze the overall weight distribution, we compute the average attention weight distribution over all test instances. However, since hidden state sequences vary in length depending on input audio duration, we interpolate the attention weight sequences to a fixed length of 100 elements. Figure 1 shows the average distribution of these resampled attention weights for all considered probes.

In probes trained on the enc-dec, Seamless pre-ad, and ZeroSwot pre-ad/post-ad representations, we observe an initial peak in attention followed by a descending trend. However, Seamless pre-ad shows a sharp peak at the very beginning of the sequence, suggesting that only the earliest portion is primarily used for classification. In contrast, for Seamless post-ad, where the probe performs worst, the initial positions appear to be the least relevant, although no clear pattern emerges. This could be due to the absence of gender encoding, leading to suboptimal probe performance.

Overall, our findings suggest that the earliest

	# Samples			# Hours			# Speakers		
	All	She	He	All	She	He	All	She	He
train	5061	2514	2546	9.70	5.08	4.62	1417	534	883
dev	1078	539	539	2.08	1.10	0.98	337	132	205
test-generic	3702	1026	2676	6.69	1.89	4.80	38	11	27
test-speaker (en→es)	555	271	284	1.33	0.69	0.64	202	104	98
test-speaker (en→fr)	595	304	291	1.36	0.71	0.65	189	88	101
test-speaker (en→it)	551	271	280	1.31	0.68	0.63	147	69	78

Table 5: Statistics of number of samples, hours, and speaker for the data splits used.



(a) test-generic

(b) test-speaker (average across language pairs)

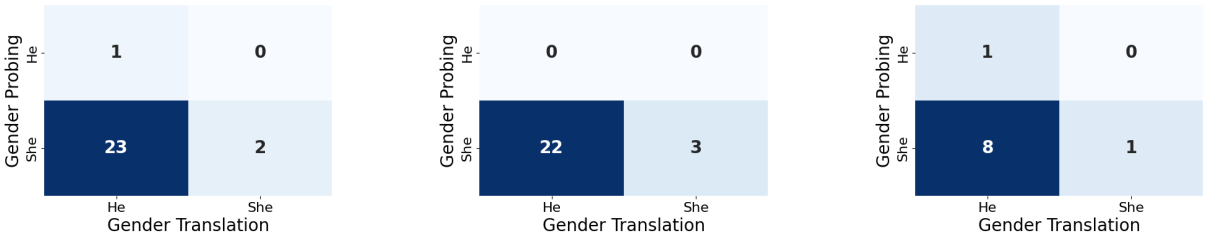
Figure 2: Mean attention weights with standard deviations for the various attention-based probes.

positions in the sequence are particularly important for predicting gender, especially when the probe is highly effective. This aligns with previous work by Krishnan et al. (2024) and Attanasio et al. (2024), who found that gender encoding is predominantly concentrated at the beginning of the sequence. However, it is important to note that the specific positions attended to can vary depending on the probe, which may develop ineffective strategies, especially when performance is suboptimal.

F A Closer Look at Divergencies Between Gender Probing and Translation

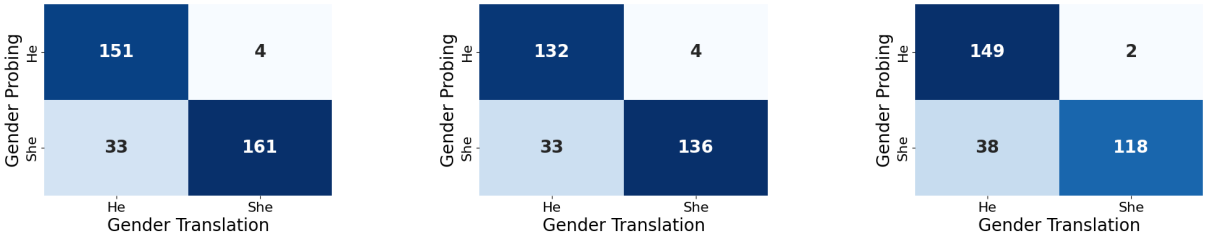
In §4.2, we show that gender encoding correlates with the model's gender translation ability. For enc-dec, both gender encoding and gender translation achieve good accuracy, although the probe's accuracy is higher. As a result, some translations are incorrect even when gender can be accurately predicted from the hidden states.

To better understand these discrepancies, we plot confusion matrices. Figure 3 shows the confusion matrix comparing *incorrect* probe predictions with the corresponding translations by the enc-dec model, while Figure 4 displays the confusion matrix for *correct* probe predictions alongside their associated translations. First, we observe that in Figure 3, the number of instances is low, as the probes are generally effective at predicting gender for enc-dec. Among the mismatches, the most frequent pattern involves the probe incorrectly predicting feminine while the model correctly translates masculine—23 cases for en→es, 22 for en→fr, and 8 for en→it. In contrast, Figure 4 shows that most discrepancies occur when the ST model translates as masculine despite the probe correctly predicting She: 33 cases for both en→es and en→fr, and 38 for en→it. The reverse case—feminine translations



(a) en-es (b) en-fr (c) en-it

Figure 3: Confusion matrices showing the relationship between *incorrect* probe predictions (gender probing) and corresponding gender translation.



(a) en-es (b) en-fr (c) en-it

Figure 4: Confusion matrices showing the relationship between *correct* probe predictions (gender probing) and corresponding gender translation.

a.	en-es	She	SRC	They have formed me as a democratic citizen and a **bridge builder**.
			REF	Me han formado como ciudadana democrática y **constructora de puentes**.
			OUT	Me han formado como ciudadana democrática y como **un constructor de puente**.
b.	en-es	He	SRC	And just as the woman who wanted to know me as **an adult** got to know me, she turned into a [...].
			REF	Y así como la mujer que quería conocerme como **adulto** llegó a conocerme, se convirtió en una [...].
			OUT	Y así como una mujer que quería conocerme como **adulta** se conoció a mí, se convirtió en una [...].
c.	en-fr	He	SRC	And I wound up getting involved with the space community, really **involved** with NASA, [...].
			REF	Et j'ai fini par être impliqué dans la communauté spatiale, réellement **impliqué** avec la NASA, [...].
			OUT	Et j'ai fini par m'impliquer dans la communauté spatiale, vraiment **impliquée** par la NASA, [...].
d.	en-fr	She	SRC	My main sport was soccer, and I was **a goalkeeper**, which is [...].
			REF	Mon sport principal était le football et j'étais **gardienne de but**, ce qui est [...].
			OUT	Mon principal sport était le foot et j'étais **un gardien**, qui est [...].
e.	en-it	She	SRC	And I want to say something a little bit radical for **a feminist**, and that is that I think that [...].
			REF	Vorrei dire qualcosa di abbastanza radicale per **una femminista**, cioè che credo [...].
			OUT	E voglio dire qualcosa di un po' radicale per **un femminista**, e penso che [...].
f.	en-it	She	SRC	As a researcher, **a professor** and a new parent, my goal is to [...].
			REF	Da ricercatrice, **professoressa** e adesso genitore, il mio obiettivo è [...].
			OUT	Da ricercatrice, **professore** e genitore nuovo, il mio obiettivo è [...].

Table 6: Transcribed source sentences (SRC) from `test-speaker`, accompanied by reference translations (REF) and output translations (OUT) generated by `enc-dec`. Misgendered words in OUT, along with their correct counterparts in REF and aligned terms in SRC, are shown in bold.

for the He class—is much rarer, with only 4 cases each for en→es and en→fr, and 2 for en→it. This asymmetry reflects the persistent gender bias in translation, which affects even the `enc-dec` model despite its overall strong performance on gender translation.

To investigate why the ST model produces incorrect translations despite correctly encoding gender, we manually examined the divergent cases from Figure 4. We identified several patterns where the model struggles, and provide examples in Table 6. In certain cases, stereotypical associations—particularly involving professions—appear to influence the translations. For instance, in *a)*, *d)*, and *f)* terms like "bridge builder", "goalkeeper", and "professor"—often associated with men—are trans-

lated as masculine even when referring to women, despite other gendered words in the sentence being correctly translated as feminine. In other cases, the model appears to struggle with complex agreement resolutions. For example, in *c)*, "involved" may have been linked to "space community" due to proximity, rather than the speaker. In *e)*, "for a feminist" may not have been correctly interpreted as referring to the speaker. In *b)*, the presence of another gendered referent ("woman" and "she") may have led the model to misinterpret "adult" as feminine.

Overall, these findings suggest that linguistic biases can sometimes override acoustic cues, limiting the model's ability to fully leverage gender information. However, a more systematic analysis is needed in future work, which is beyond the scope of this paper.

Rethinking Semantic Parsing for Large Language Models: Enhancing LLM Performance with Semantic Hints

Kaikai An[1,2]*, Shuzheng Si[1,2]*, Helan Hu[1,2], Haozhe Zhao[1,2], Yuchi Wang[1], Qingyan Guo[3], Baobao Chang[1]†
[1] National Key Laboratory for Multimedia Information Processing, Peking University
[2] School of Software and Microelectronics, Peking University [3] Tsinghua University
ankaikai@stu.pku.edu.cn, chbb@pku.edu.cn

Abstract

Semantic Parsing aims to capture the meaning of a sentence and convert it into a logical, structured form. Previous studies show that semantic parsing enhances the performance of smaller models (e.g., BERT) on downstream tasks. However, it remains unclear whether the improvements extend similarly to LLMs. In this paper, our empirical findings reveal that, unlike smaller models, directly adding semantic parsing results into LLMs reduces their performance. To overcome this, we propose SENSE, a novel prompting approach that embeds semantic hints within the prompt. Experiments show that SENSE consistently improves LLMs' performance across various tasks, highlighting the potential of integrating semantic information to improve LLM capabilities.

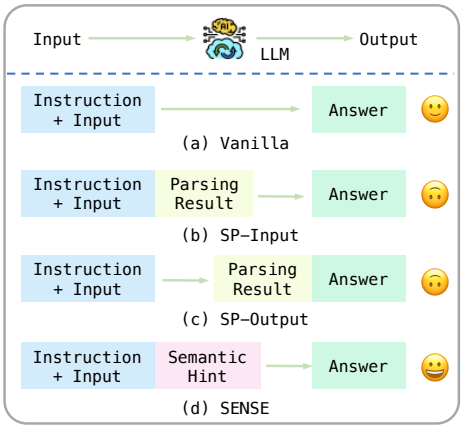


Figure 1: Different ways of evaluating LLMs on downstream tasks. While (a) represents direct prompting models, (b) and (c) add semantic parsing results either from the input or output side. The upside-down face indicates a negative impact. Our method, SENSE, introduces semantic hints without perception of the results.

1 Introduction

Semantic Parsing is a fundamental task in Natural Language Processing, which involves converting a natural language sentence into structured meaning representation. This includes tasks like Semantic Role Labeling (SRL), Frame Semantic Parsing (FSP) and Abstract Meaning Representation (AMR) (Gildea and Jurafsky, 2002; Baker et al., 2007; Banarescu et al., 2013; Palmer et al., 2010; An et al., 2023). Such structured information are applicable across various tasks, like Question Answering (Khashabi et al., 2022), Machine Translation (Rapp, 2022), Dialogue Systems (Xu et al., 2020; Si et al., 2022, 2024) and so on.

Previous work from Bonial et al. (2020); Rapp (2022); Khashabi et al. (2022) demonstrate that integrating semantic parsing results from SRL or AMR parsing into a model's input can effectively enhance its ability to understand illocutionary acts and linguistic abstractions, thereby improving downstream performance. However, these findings are largely limited to smaller models like BERT (Devlin et al., 2019). With the rise of Large Language Models (LLMs), it becomes essential to explore how the integration of semantic parsing could impact. Recently, Jin et al. (2024) investigates the role of semantic representation in LLMs by proposing AMRCOT, a method similar to that depicted in Fig.1 (b). Their findings reveal that introducing AMR results into the input generally harms LLM performance more than it helps, likely because AMR is not yet a representation well-suited for LLMs. However, this analysis remains limited, as it only considers the effects of AMR on several tasks, leaving the broader potential of semantic parsing in LLMs largely unexplored.

In this paper, we systematically investigate the impact of semantic parsing on LLMs to address the question: ***Can Semantic Information Still Contribute to Improve Downstream Tasks on LLMs?*** We empirically compare different paradigms for integrating semantic parsing into LLMs, as shown in Fig.1. These paradigms include approaches com-

* Equal contribution
† Corresponding author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1020–1029
July 27 - August 1, 2025

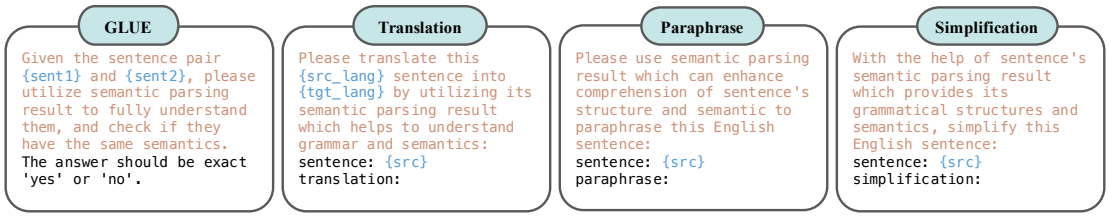


Figure 2: Illustration of SENSE designed for downstream tasks.

monly used for smaller models, such as incorporating semantic parsing results directly on the input side by fine-tuning or integrating them on the output side. However, these methods negatively affect model performance since they limit fixed types of semantic parsing and might introduce erroneous results. Thus, we propose a novel prompting approach, **SENSE**, illustrated in Fig.1 (d). Instead of injecting explicit parsing results, SENSE encourages LLMs to harness their internal semantic parsing capabilities through the addition of semantic hints. These hints are as simple as ***"please use semantic parsing result to enhance comprehension of the sentence's structure and semantics"***. Our comprehensive experiments demonstrate that SENSE promote LLM to focus more on key semantic information, not only achieves superior and consistent performance across various tasks, but also produces more linguistically aligned results, particularly on simplification and paraphrasing tasks, underscoring the effectiveness of semantic parsing for enhancing LLMs' performance.

2 Semantic Information → LLMs

In this section, we delve into answering the question: ***Can Semantic Information Still Contribute to Improve Downstream Tasks on LLMs?***

2.1 Methodology

Previous studies, such as those by Ettinger et al. (2023) and Jin et al. (2024), highlight the difficulty LLMs face in processing the schemes and symbols of explicit semantic parsing results. Their findings suggest that directly integrating these results can degrade model performance. Given that LLMs are already capable of achieving strong results in an end-to-end manner, we propose a novel approach: incorporating semantic parsing hints into the instruction to prompt LLMs to leverage their internal parsing capabilities.

As Fig.2 shows, our SENSE introduces simple semantic hints such as *"utilize semantic parsing result"* to *"fully understand input"* or *"capture grammatical structures and semantics"* to complete downstream tasks. This strategy encourages LLMs to engage in inherent understanding of linguistic structures without requiring explicit semantic parsing results. The workflow outlined in Fig.1 (d) demonstrates how semantic hints are integrated, and SENSE works in an zero-shot manner.

2.2 Datasets and Evaluation

In our experiments, we select seven understanding tasks from GLUE and three representative generation tasks including Machine Translation, Paraphrasing, and Simplification. Specifically, for paraphrasing task, we report three linguistic metrics across lexical, syntactic, and semantic levels, for simplification task, we report SARI and SAMSA which evaluate the predicted simplified sentences from lexical structure and semantic meaning preservation. More details about our experiments can be found in Appendix A.1 and A.2.

3 Experimental Results

3.1 Main Results

Results on Understanding Tasks From Table 1, the results demonstrate that although LLMs currently lag behind smaller models like BERT, the integration of SENSE significantly narrows this gap. Specifically, SENSE improves the average performance of GPT-4o-mini from 79.43% to 81.25%, bringing it closer to BERT's performance of 83.2%. Moreover, SENSE is effective in enhancing the performance of both closed-source models such as GPT-series, and open-source models like LLaMA3. Across all GLUE tasks, SENSE consistently yields performance gains, with notable improvements in MRPC (72.30% to 76.47%), MNLI (73.90% to 78.20%) and CoLA (65.49% to 67.22%). These results highlight SENSE's ability to enhance LLMs' comprehension of input sentences and demonstrate its robustness across diverse tasks.

Results on Paraphrasing Table 2 indicates that SENSE effectively enhances linguistic diversity in paraphrasing tasks while maintaining high semantic similarity. Notably, SENSE retains the

System	SST-2	MRPC	QQP	MNLI	QNLI	RTE	CoLA	Average
	Acc	Acc	Acc	Acc	Acc	Acc	Mcc	
$BERT_{LARGE}$ (2019)	93.20	88.00	91.30	86.60	92.30	70.40	60.60	83.20
$RoBERTa_{LARGE}$ (2019)	96.40	90.90	92.20	90.20	94.70	86.60	68.00	88.43
LLaMA3-70B	**95.64**	73.52	74.60	71.90	91.30	84.48	63.90	79.34
+ SENSE	95.18	74.04	76.50	73.10	92.80	85.56	65.53	80.25
GPT-3.5-turbo	91.86	73.28	73.40	61.80	82.40	81.81	63.50	75.44
+ SENSE	92.20	75.49	**77.20**	64.60	83.20	84.12	64.57	77.34
GPT-4o-mini	91.63	72.30	73.00	73.90	92.30	87.36	65.49	79.43
+ SENSE	92.08	**76.47**	73.00	**78.20**	**93.30**	**88.45**	**67.22**	**81.25**

Table 1: Experimental results on GLUE benchmark.

System	Prediction–Source		
	Semantic Similarity ↑	Lexical Overlap ↓	Syntactic Diversity ↑
LLaMA3-70B	83.71	30.00	10.85
+ SENSE	**84.02**	**29.00**	**11.51**
GPT-3.5-turbo	85.79	46.37	8.76
+ SENSE	**85.79**	**25.33**	**10.24**
GPT-4o-mini	89.71	39.00	7.25
+ SENSE	**90.26**	**34.00**	**8.08**

Table 2: Experimental results on Paraphrasing. We report linguistic metrics between source and prediction.

semantic similarity score at 90.26 but significantly reduces lexical overlap from 39.00 to 34.00 and increases syntactic diversity from 7.25 to 8.08. This indicates that the semantic hints introduced by SENSE lead to more diverse syntactic structures and reduced lexical repetition while preserving the core meaning of the source sentence, which validates the effectiveness of SENSE in generating paraphrases that are not only semantically faithful but also exhibit greater lexical and syntactic variety.

System	BLEU ↑	SARI ↑	SAMSA ↑
TrukCorpus			
GPT-3.5-turbo	58.16	42.25	31.42
+ SENSE	**63.42**	**42.42**	**37.03**
GoogleComp			
GPT-3.5-turbo	13.12	35.53	28.14
+ SENSE	**14.31**	**35.67**	**30.52**

Table 3: Experimental results on Simplification. We add two metrics, SARI and SAMSA to evaluate the semantic structure of the output.

Results on Simplification Table 3 showcases the improved performance of SENSE on two simplification datasets. Compared to the vanilla prompt, SENSE delivers higher BLEU scores of 63.42 on TrukCorpus and 14.31 on GoogleComp, alongside a modest increase in SARI, which evaluates the alignment between the source and target sentences. More importantly, the SAMSA scores, which measure the preservation of syntactic structure, show substantial improvement, reaching 37.03 and 30.52 respectively. These results demonstrate that integrating semantic hints into prompts enhances the model's ability to simplify sentences while preserving their original structure, resulting in more effective overall simplification.

Results on Machine Translation We further conduct experiments on Machine Translation task and present a comparative analysis of GPT-3.5-turbo across the vanilla prompt, our SENSE, and other state-of-the-art systems in Table 8. Results show that SENSE consistently enhances GPT-3.5 across all evaluated metrics and language pairs. For the DE-EN task, SENSE achieves the highest scores: COMET22 (86.44), ChrF (59.08), and BLEU (33.75), outperforming the WMT-Best system. Similarly, in the EN-DE task, SENSE significantly boosts GPT-3.5's performance, reaching COMET22 (86.65), ChrF (62.84), and BLEU (34.18). These improvements highlight the effectiveness of SENSE in enhancing GPT-3.5's ability to handle translation tasks across different language pairs. The results for ZH-EN and EN-ZH in Table 8 further confirm SENSE's effectiveness.

3.2 Analytical Results

Analysis of Different Paradigms In Table 4, we compare various approaches for incorporating semantic parsing into LLMs. We examine methods that either concatenate pre-generated parsing results using LLM or generate them on output side[1]. The results demonstrate that directly adding semantic parsing results degrades performance, aligning with findings by Jin et al. (2024). This degradation arises from the unfamiliar symbolic representation and the diversity of semantic parsing tasks, integrating specific type, and potentially erroneous results limits LLM's capability. In contrast, SENSE avoids explicit incorporation while consistently outperforming these methods. Such finding un-

[1]We do not specify certain type of semantic parsing during our experiments.

System	SST-2	MRPC	QQP	MNLI	QNLI	RTE	CoLA
GPT-3.5-turbo	91.86	73.28	73.40	61.80	82.40	81.81	63.50
+ CoT (2022)	$89.11_{-2.75}$	$73.28_{+0.00}$	$77.00_{+3.60}$	$56.20_{-5.60}$	$82.70_{+0.30}$	$82.54_{+0.73}$	$64.32_{+0.82}$
+ SP-Input	$87.50_{-4.36}$	$74.26_{+0.98}$	$74.30_{+0.90}$	$50.50_{-11.30}$	$78.40_{-4.00}$	$84.11_{+2.30}$	$58.37_{-5.13}$
+ SP-Output	$89.11_{-2.75}$	$73.52_{+0.24}$	$71.90_{-1.50}$	$62.00_{+0.20}$	$78.40_{-4.00}$	$81.59_{-0.22}$	$64.44_{+0.94}$
+ SENSE	$92.20_{+0.34}$	$75.49_{+2.21}$	$77.20_{+3.80}$	$64.60_{+2.80}$	$83.20_{+0.80}$	$84.12_{+2.31}$	$64.57_{+1.07}$

Table 4: Analysis of different approaches that introduce semantic parsing into LLMs on GLUE benchmark. Improvements are marked in red and decreases in green, relative to GPT-3.5-turbo.

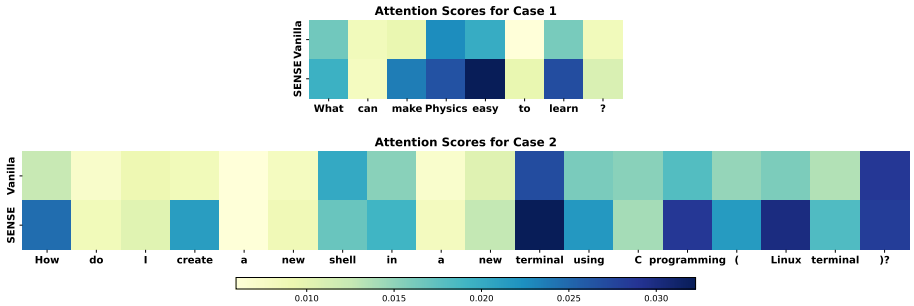


Figure 3: Visualization of attention scores from LLaMA3-70B on the source sentence in the Paraphrasing Task.

derscores SENSE as a more effective strategy for leveraging semantic parsing on LLMs.

Comparison with Chain-of-Thought Since SENSE shares similarities with CoT (Kojima et al., 2022), which works by adding "Let's think step by step", we compare it on GLUE (Table 4) and machine translation task (Table 8). While CoT degrades performance across tasks, as it is better suited for reasoning tasks, SENSE significantly enhances LLM performance by improving the model's ability to understand input sentences, thus yielding better results.

Visualization of Attention Scores We present the distribution of attention scores for paraphrasing task in Fig.3, where we average attention scores for each output token with respect to original sentence. The visualization reveals that, compared to vanilla prompt, SENSE places greater emphasis on key semantic elements, such as important lexical units and core components. This indicates that SENSE more effectively directs attention toward critical semantic information, and thus generates outputs that are more linguistic-aligned. Additionally, we provide case study on such examples in Table 9 and 10. While both vanilla prompt and SENSE successfully capture the paraphrased meaning, SENSE is superior at transforming syntactical structure and utilizing more diverse expressions.

4 Related Work

Semantic parsing has significantly contributed to enhancing the performance of smaller language models. Integrating results from SRL and AMR (Gildea and Jurafsky, 2002; Palmer et al., 2010; Banarescu et al., 2013) has shown to improve model performance on various tasks (Khashabi et al., 2022; Rapp, 2022; Xu et al., 2020; Si et al., 2022, 2024). However, the effectiveness of semantic parsing to LLMs is under-explored. Recent work, such as Jin et al. (2024), explores the use of AMR results with LLMs and finds that direct integration of these results may not always yield positive influences. Unlike approaches focused on optimizing prompts directly (Zhou et al., 2022; Pryzant et al., 2023; Deng et al., 2022; Guo et al., 2024), our work proposes a novel strategy for leveraging semantic parsing in LLMs. Similar to CoT (Wei et al., 2022; Kojima et al., 2022) and DTG (Li et al., 2023), our method involves integrating semantic parsing hints into prompts rather than optimizing the prompts.

5 Conclusion

In this paper, we rethink leveraging semantic parsing to enhance LLMs' performance. Contrary to smaller models, where direct integration of parsing results can be beneficial, we find that this negatively impacts LLMs. With the help of our proposed SENSE, which introduces semantic hints within prompts, LLMs can better comprehend input sentences. Experiments show that SENSE achieves great performance across both understanding and generation tasks, and helps models capture lexical and syntactic structures, producing outputs that align more closely with linguistic metrics.

Limitations

While we validate the effectiveness of SENSE across both understanding and generation tasks, there are limitations that remain for future exploration: Firstly, our validation is restricted to the LLaMA and GPT-series models. Extending SENSE to other LLM architectures will be necessary to confirm its general applicability. Secondly, although SENSE shows promising results on a range of NLP tasks, its performance across more diverse datasets and applications needs further investigation. Our experiments focus on tasks where the benefits of semantic parsing have been established, but broader testing is required to fully assess its potential. Additionally, the underlying mechanism of how semantic parsing influences LLM decision-making remains unclear, as LLMs function largely as black-box systems. Our validation primarily involves comparing methods that directly incorporate semantic parsing results from the input or output sides, and analyzing the outputs in contrast to both the vanilla prompt and SENSE.

Acknowledgements

We thank all reviewers for their great efforts. This work is supported by the National Science Foundation of China under Grant No.61876004 and 61936012.

References

Kaikai An, Ce Zheng, Bofei Gao, Haozhe Zhao, and Baobao Chang. 2023. Coarse-to-fine dual encoders are better frame identification learners. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13455–13466, Singapore. Association for Computational Linguistics.

Collin Baker, Michael Ellsworth, and Katrin Erk. 2007. SemEval-2007 task 19: Frame semantic structure extraction. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 99–104, Prague, Czech Republic. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. *Linguistic Annotation Workshop,Linguistic Annotation Workshop*.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-amr: abstract meaning representation for dialogue. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 684–695.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P Xing, and Zhiting Hu. 2022. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *arXiv preprint arXiv:2205.12548*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Allyson Ettinger, Jena Hwang, Valentina Pyatkin, Chandra Bhagavatula, and Yejin Choi. 2023. "you are an expert linguistic annotator": Limits of LLMs as analyzers of Abstract Meaning Representation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8250–8263, Singapore. Association for Computational Linguistics.

Daniel Gildea and Dan Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2024. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *Preprint*, arXiv:2309.08532.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*.

Kuan-Hao Huang, Varun Iyer, I-Hung Hsu, Anoop Kumar, Kai-Wei Chang, and Aram Galstyan. 2023. ParaAMR: A large-scale syntactically diverse paraphrase dataset by AMR back-translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8047–8061, Toronto, Canada. Association for Computational Linguistics.

Zhijing Jin, Yuen Chen, Fernando Gonzalez, Jiarui Liu, Jiayi Zhang, Julian Michael, Bernhard Schölkopf, and Mona Diab. 2024. Analyzing the role of semantic representations in the era of large language models. *arXiv preprint arXiv:2405.01502*.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2022. Question answering as global reasoning over semantic abstractions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Bei Li, Rui Wang, Junliang Guo, Kaitao Song, Xu Tan, Hany Hassan, Arul Menezes, Tong Xiao, Jiang Bian, and JingBo Zhu. 2023. Deliberate then generate: Enhanced prompting framework for text generation. *arXiv preprint arXiv:2305.19835*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

OpenAI. 2023. Chatgpt: Optimizing language models for dialogue. https://openai.com/blog/chatgpt. Accessed: 2023-04-01.

Martha Palmer, Ivan Titov, and Shumin Wu. 2010. Semantic role labeling. *Computational Linguistics*.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*.

Reinhard Rapp. 2022. Using semantic role labeling to improve neural machine translation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3079–3083.

Ricardo Rei, José GC De Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André FT Martins. 2022. Comet-22: Unbabel-ist 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

Shuzheng Si, Wentao Ma, Haoyu Gao, Yuchuan Wu, Ting-En Lin, Yinpei Dai, Hangyu Li, Rui Yan, Fei Huang, and Yongbin Li. 2024. Spokenwoz: A large-scale speech-text benchmark for spoken task-oriented dialogue agents. *Advances in Neural Information Processing Systems*, 36.

Shuzheng Si, Shuang Zeng, and Baobao Chang. 2022. Mining clues from incomplete utterance: A query-enhanced network for incomplete utterance rewriting. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4839–4847, Seattle, United States. Association for Computational Linguistics.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. Semantic structural evaluation for text simplification. *arXiv preprint arXiv:1810.05022*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Preprint*, arXiv:1804.07461.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Kun Xu, Haochen Tan, Linfeng Song, Han Wu, Haisong Zhang, Linqi Song, and Dong Yu. 2020. Semantic role labeling guided multi-turn dialogue rewriter. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6632–6639.

Shaolei Zhang, Qingkai Fang, Zhuocheng Zhang, Zhengrui Ma, Yan Zhou, Langlin Huang, Mengyu Bu, Shangtong Gui, Yunji Chen, Xilin Chen, et al. 2023. Bayling: Bridging cross-lingual alignment and instruction following through interactive translation for large language models. *arXiv preprint arXiv:2306.10968*.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

A Supplementary Details

A.1 Details about Datasets

We list the details of each dataset, including source, number, and metrics for each task in Table 5, and we sample a subset of data if the original dataset is large to reduce the API cost.

Dataset	Num.	Metrics
SST-2	872	Acc
MRPC	408	Acc
QQP	1000	Acc
MNLI	1000	Acc
QNLI	1000	Acc
RTE	277	Acc
CoLA	1053	Mcc
WMT DE-EN	1984	BLEU, COMET22, Chrf
WMT EN-DE	1875	BLEU, COMET22, Chrf
WMT ZH-EN	1875	BLEU, COMET22, Chrf
WMT EN-ZH	1875	BLEU, COMET22, Chrf
QQP	2500	Lexical, Syntactic, Semantic
TurkCorpus	359	BLEU, SARI, SAMSA
GoogleComp	1000	BLEU, SARI, SAMSA

Table 5: Statistics of the dataset we use in our experiment.

GLUE We test on seven tasks from GLUE benchmark (Wang et al., 2019) and report the Matthews Correlation Coefficient (MCC) for CoLA and Accuracy (Acc) for the left tasks.

Machine Translation For machine translation, we evaluate our method on the WMT22 [2] dataset, focusing on two language pairs: EN-DE (English to German) EN-ZH (English to Chinese) and report COMET22 (Rei et al., 2022), CHRF, and BLEU scores [3].

Paraphrasing We evaluate on Quora Question Pairs (QQP) [4] dataset. To analyze results professionally, we follow Huang et al. (2023) and report three linguistic evaluation metrics across lexical, syntactic, and semantic levels.

Simplification For text simplification, we evaluate on TurkCorpus and GoogleComp and use BLEU, SARI, and SAMSA as the evaluation metrics. Specifically, SARI [5] (System output Against References and against the Input sentence) is used to compare the predicted simplified sentences against the reference and the source sentences, and SAMSA (Sulem et al., 2018) is a metric specifically designed for text simplification that evaluates structural simplification and meaning preservation.

A.2 Details about Experiment

A.2.1 Experimental Setup

We test our SENSE on GPT-3.5-turbo, GPT-4o-mini (OpenAI, 2023) with the version of 2023-11-06 and 2024-07-18, and LLaMA3-70B-Instruct [6]. The temperature is set to 0 and top_p set to 1.

A.2.2 Prompts used in Experiments

We release the prompts we use during our experiments in Table 6 and Table 7.

A.3 Additional Experimental Results

Results on WMT22 From Table 8, for the ZH-EN translation task, SENSE improves GPT-3.5-turbo's ChrF (58.50) and BLEU (27.04) scores, though the COMET22 score (80.47) is slightly lower than the baseline. In the EN-ZH task, SENSE achieves the highest COMET22 (88.06) and enhances ChrF (39.86) and BLEU (44.40) compared to baselines.

Case Study In Tables 9 and 10, we present case studies on paraphrasing and inference tasks. These demonstrate that SENSE not only excels in altering syntactic structures and employing a broader range of expressions, thereby enhancing the overall quality of paraphrasing, but also better captures sentence semantics.

[2] https://machinetranslate.org/wmt22

[3] BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a

[4] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

[5] https://huggingface.co/spaces/evaluate-metric/sari

[6] https://llama.meta.com/docs/model-cards-and-prompt-formats/meta-llama-3

Dataset	Method	Prompt
SST-2	Vanilla	Given this sentence: {sentence}, please classify its sentiment as positive or negative. The answer should be exactly 'positive' or 'negative'.
	CoT	Given this sentence: {sentence}, please think step by step, and then classify its sentiment as positive or negative. The answer should be exactly 'positive' or 'negative'.
	SP-Input	Given this sentence: {sentence} and its semantic parsing result {parsing}, please classify the sentence's sentiment as positive or negative. The answer should be exactly 'positive' or 'negative'.
	SP-Output	Given this sentence: {sentence}, please first parse this sentence and then classify the sentence's sentiment as positive or negative. The answer should be exactly 'positive' or 'negative'.
	SENSE	Given this sentence: {sentence}, please use semantic parsing result which can enhance comprehension of the sentence's structure and semantics to classify the sentence's sentiment. The answer should be exactly 'positive' or 'negative'.
MRPC	Vanilla	Given the sentence pair {sentence1} and {sentence2}, please check if these two sentences have the same semantics. The answer should be exactly 'yes' or 'no'.
	CoT	Given the sentence pair {sentence1} and {sentence2}, please think step by step, and then check if these two sentences have the same semantics. The answer should be exactly 'yes' or 'no'.
	SP-Input	Given the sentence pair {sentence1} and {sentence2} and their semantic parsing results {parsing1} and {parsing2}, please check if these two sentences have the same semantics. The answer should be exactly 'yes' or 'no'.
	SP-Output	Given the sentence pair {sentence1} and {sentence2}, please first parse these sentences and then check if these two sentences have the same semantics. The answer should be exactly 'yes' or 'no'.
	SENSE	Given the sentence pair {sentence1} and {sentence2}, please use semantic parsing result which can enhance comprehension of the sentence's structure and semantics to measure if these two sentences have the same semantics. The answer should be exactly 'yes' or 'no'.
MNLI	Vanilla	Given the sentence1 {premise} and sentence2 {hypothesis}, determine whether sentence2 entail, contradict, or is it neutral to sentence1. The answer should be exactly 'entail' or 'contradict' or 'neutral'.
	CoT	Given the sentence1 {premise} and sentence2 {hypothesis}, please think step by step, and then determine whether sentence2 entail, contradict, or is it neutral to sentence1. The answer should be exactly 'entail' or 'contradict' or 'neutral'.
	SP-Input	Given the sentence1 {premise} and sentence2 {hypothesis} and their semantic parsing results {parsing1} and {parsing2}, please determine whether sentence2 entail, contradict, or is it neutral to sentence1. The answer should be exactly 'entail' or 'contradict' or 'neutral'.
	SP-Output	Given the sentence1 {premise} and sentence2 {hypothesis}, please first parse these sentence to fully understand its structure and semantics and then determine whether sentence1 entail, contradict, or is neutral to sentence2. The answer should be exactly 'entail' or 'contradict' or 'neutral'.
	SENSE	Given the sentence1 {premise} and sentence2 {hypothesis}, please use semantic parsing result which can enhance comprehension of the sentence's structure and semantics to determine whether sentence1 entail, contradict, or is neutral to sentence2. The answer should be exactly 'entail' or 'contradict' or 'neutral'.
QNLI	Vanilla	Given the sentence1 {question} and sentence2 {sentence}, please determine if the sentence contains the answer to the question. The answer should be exactly 'entail' or 'not entail'.
	CoT	Given the sentence1 {question} and sentence2 {sentence}, please think step by step, and then determine if the sentence contains the answer to the question. The answer should be exactly 'entail' or 'not entail'.
	SP-Input	Given the sentence1 {question} and sentence2 {sentence} and their semantic parsing results {parsing1} and {parsing2}, please determine if the sentence contains the answer to the question. The answer should be exactly 'entail' or 'not entail'.
	SP-Output	Given the sentence1 {question} and sentence2 {sentence}, please first parse these sentences and then determine if the sentence contains the answer to the question. The answer should be exactly 'entail' or 'not entail'.
	SENSE	Given the sentence1 {question} and sentence2 {sentence}, please use semantic parsing result which can enhance comprehension of the sentence's structure and semantics to determine if the sentence contains the answer to the question. The answer should be exactly 'entail' or 'not entail'.
CoLA	Vanilla	Given the sentence: {sentence}, please check if the sentence is grammatically correct. The answer should be exactly 'yes' or 'no'.
	CoT	Given the sentence: {sentence}, please think step by step, and then check if the sentence is grammatically correct. The answer should be exactly 'yes' or 'no'.
	SP-Input	Given the sentence: {sentence} and its semantic parsing result {parsing}, please check if the sentence is grammatically correct. The answer should be exactly 'yes' or 'no'.
	SP-Output	Given the sentence: {sentence}, please first parse this sentence and then check if the sentence is grammatically correct. The answer should be exactly 'yes' or 'no'.
	SENSE	Given the sentence: {sentence}, please use semantic parsing result which can enhance comprehension of the sentence's structure and semantics to check if the sentence is grammatically correct. The answer should be exactly 'yes' or 'no'.

Table 6: We list the prompts we use during our experiments on GLUE benchmarks and omit QQP and RTE since QQP is similar to MRPC and RTE is similar to MNLI.

Dataset	Method	Prompt
WMT22	Vanilla	Please translate this {src_lang} sentence into {tgt_lang}: sentence: {src} translation:
	SENSE	Please translate this {src_lang} sentence into {tgt_lang} by utilizing its semantic parsing result which helps to understand grammar and semantics: sentence: {src} translation:
Simplification	Vanilla	Please simplify this English sentence: sentence: {src} simplification:
	SENSE	With the help of the sentence's semantic parsing result which provides its grammatical structures and semantics, simplify this English sentence: sentence: {src} simplification:
Paraphrasing	Vanilla	Please paraphrase this English sentence: sentence: {src} paraphrase:
	SENSE	Please use semantic parsing result which can enhance comprehension of sentence's structure and semantic to paraphrase this English sentence: sentence: {src} paraphrase:

Table 7: We list the prompts we use during our experiments on generation tasks.

System	DE-EN			EN-DE		
	COMET22 ↑	ChrF ↑	BLEU ↑	COMET22 ↑	Chrf ↑	BLEU ↑
WMT-Best	85.00	58.50	33.40	87.20	64.60	38.40
GPT EVAL (2023)	84.80	58.30	33.40	84.20	59.60	30.90
DTG 5-shot (2023)	85.40	58.20	33.20	86.30	61.60	33.40
BayLing (2023)	85.47	58.65	32.94	**86.93**	62.76	34.12
GPT-3.5-turbo	85.71	58.19	33.15	84.60	60.48	33.42
+ CoT	84.99	57.74	31.46	84.95	61.17	29.70
+ SENSE	**86.44**	**59.08**	**33.75**	86.65	**62.84**	**34.18**
System	ZH-EN			EN-ZH		
	COMET22 ↑	ChrF ↑	BLEU ↑	COMET22 ↑	Chrf ↑	BLEU ↑
WMTBest	81.00	61.10	33.50	86.70	41.10	44.80
GPT EVAL (2023)	81.20	56.00	25.90	84.40	36.00	40.30
DTG 5-shot (2023)	81.70	55.90	25.20	86.60	39.40	43.50
BayLing (2023)	**82.64**	57.90	26.13	86.81	**40.32**	**44.99**
GPT-3.5-turbo	80.60	58.40	26.93	81.48	37.80	42.85
+ SENSE	80.47	**58.50**	**27.04**	**88.06**	39.86	44.40

Table 8: Experimental results on WMT22.

Source: What can make Physics easy to learn?
Target: How can you make physics easy to learn?
Vanilla Prediction: What makes learning Physics easier?
SENSE Prediction: What factors can help simplify the learning of Physics?

Source: How do I create a new shell in a new terminal using C programming (Linux terminal)?
Target: How do I create a new Terminal and new shell in Linux using C programming?
Vanilla Prediction: How do I open a new shell in a new terminal window using C programming in a Linux environment?
SENSE Prediction: What steps should I follow to launch a new shell in a separate terminal using C programming on a Linux system?

Table 9: Case study comparing SENSE and Vanilla prompt predictions for the paraphrasing task.

Premise: My article does not say or imply that real earnings growth only reflects retentions and that dividend growth must be zero or that all valuation techniques are out the window for firms that don't pay dividends. **Hypothesis:** My article doesn't say or imply that real earnings growth reflects only retentions and that dividend growth must be zero or that valuation techniques are unused for firms which don't pay dividends, it implies that growth is a concrete and calculable statistic. **Ground Truth:** Neutral **Vanilla Prediction:** Contradict **SENSE Prediction:** The semantic parsing result of both sentences indicates that they are expressing the same idea, with some minor differences in wording. Therefore, the two sentences are neutral to each other. **Analysis:** In this case, SENSE correctly identifies that both sentences express the same idea with minor wording differences, thus labeling them as neutral. The baseline prediction incorrectly identifies a contradiction.
Premise: Is there a favorite story or storybook or anything that you had when you were little that somebody read to you? **Hypothesis:** You don't remember your childhood, do you? **Ground Truth:** Contradiction **Vanilla Prediction:** Sentence1 entails sentence2. If someone is asking about a favorite story or storybook from childhood, it implies that they believe the person has memories of being read to as a child. **SENSE Prediction:** Based on the semantic parsing result, sentence1 is neutral to sentence2. The first sentence is asking about a favorite story from childhood, while the second sentence is questioning the person's memory of their childhood. There is no direct contradiction or entailment between the two sentences. **Analysis:** SENSE detects that the first sentence is asking about a favorite story from childhood, while the second sentence is questioning the person's memory of their childhood. Since it regards the second sentence as a question without an answer, it believes that there is no direct contraction or entailment, but regardless, this example should not be the implied relationship as entailment predicted by the baseline.

Table 10: Case study comparing SENSE and Vanilla prompt predictions for the Natural Language Inference task.

Quantifying Misattribution Unfairness in Authorship Attribution

Pegah Alipoormolabashi
Stony Brook University
palipoormola@cs.stonybrook.edu

Ajay Patel
University of Pennsylvania
ajayp@seas.upenn.edu

Niranjan Balasubramanian
Stony Brook University
niranjan@cs.stonybrook.edu

Abstract

Authorship misattribution can have profound consequences in real life. In forensic settings simply being considered as one of the potential authors of an evidential piece of text or communication can result in undesirable scrutiny. This raises a fairness question: Is every author in the candidate pool at equal risk of misattribution? Standard evaluation measures for authorship attribution systems do not explicitly account for this notion of fairness. We introduce a simple measure, Misattribution Unfairness Index (MAUI_k), which is based on how often authors are ranked in the top k for texts they did *not* write. Using this measure we quantify the unfairness of five models on two different datasets. All models exhibit high levels of unfairness with increased risks for some authors. Furthermore, we find that this unfairness relates to how the models embed the authors as vectors in the latent search space. In particular, we observe that the risk of misattribution is higher for authors closer to the centroid (or center) of the embedded authors in the haystack. These results indicate the potential for harm and the need for communicating with and calibrating end users on misattribution risk when building and providing such models for downstream use.

1 Introduction

Authorship attribution has various sensitive uses in multiple domains such as literature (Zhao and Zobel, 2007; Stańczyk and Cyran, 2007) and forensics (M. et al., 2016). In some uses, authorship misattribution or even suspected authorship can have dire consequences for the individuals involved. For example, consider a scenario where a forensic investigation hinges on finding who wrote a piece of text from a large pool of potential authors. Automatic authorship attribution is often seen as a way to reduce the number of candidates for such settings (Tschuggnall et al., 2019). However, even being suspected of authorship in this setting can result in further scrutiny. Given such undesirable consequences, we ask a question of fairness: *Are some authors more likely to be misattributed than others?*

To answer this question, we study the commonly used *needle-in-the-haystack* formulation (Rivera-Soto et al., 2021; Wang et al., 2023; Wen et al., 2024). In this setting, the *haystack* is a collection of known authors and documents they have authored D_h. Given some documents D_q from a query author i.e., ones whose authorship is unknown as of yet, the attribution task is to find a candidate author (*needle*) from the haystack who most likely authored the query documents. The task is then framed as a ranking problem, where systems rank haystack authors by comparing an embedding of their documents $e_i = \text{enc}(D_h(a_i))$ to an embedding of the query documents $e_q = \text{enc}(D_q)$. Such *embed-and-rank* solutions are appealing since they allow efficient scaling to large haystacks with many authors (Douze et al., 2024).

Existing evaluation measures for these solutions do not tackle fairness notions. Prior work mostly use effectiveness metrics, such as Mean Reciprocal Rank (MRR), or Recall at various ranks (R@k), or Recall at various ranks (R@k) (Andrews and Bishop, 2019; Khan et al., 2021; Rivera-Soto et al., 2021; Burrows et al., 2009). These metrics are designed to reduce the amount of manual scrutiny by assessing whether the correct author is ranked as highly as possible. They do not capture or measure the risks of getting ranked highly for other authors.

We make two contributions to remedy this gap: **1)** We introduce a way to measure the unfairness in rankings induced by models and empirically assess unfairness across five models over two datasets. **2)** We perform an analysis and provide a potential explanation for unfairness in how embedding distributions relate to misattribution risks of authors. Our results show that model rankings exhibit high levels of unfairness and authors closer to the center of the embedding space are at a higher risk for

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1030–1041
July 27 - August 1, 2025

misattribution. These call for further research both in evaluation and in modeling of authorship attribution systems to reduce the potential for unfairness-related harms.

2 Misattribution Unfairness in Author Ranking

We introduce a notion of fairness where all users carry equal risks of being misattributed i.e., are equally likely to be ranked high for documents that they have not authored. This is similar to the common definition of fairness in retrieval settings (Biega et al., 2020). In retrieval, the focus is that all authors receive relevance-proportional attention in the rankings. Here we focus on reducing undue or disproportional presence in ranking.

Suppose there are N_h authors in the haystack and N_q query authors selected at random from the haystack. For any single query, if we consider an unbiased ranking i.e. a random permutation of the authors, the probability of any specific author being ranked higher than k is $\frac{k}{N_h}$. When we query for N_q times, an author is expected to get ranked higher than k for $E_k = \lceil \frac{k}{N_h} \times N_q \rceil$ times.

The unfairness of model-induced rankings can thus be characterized in terms of how the actual counts of authors being ranked at top k exceeds the expected count[1]. Let c_j^k denote the number of times author a_j is ranked in the top k. Then, we can quantify the unfairness as follows:

$$\text{MAUI}_\text{k} = \frac{\sum_{j=1}^{N_h} \max(0, (c_j^k - E_k))}{k \times (N_q - E_k)} \quad (1)$$

This metric normalizes the sum of the differences by its highest possible value which happens in the worst case: when the same k authors are ranked in top k for all queries. This scales the values between 0 and 1, 0 being most fair and 1 being the least.

3 Evaluation

We experiment with multiple authorship models and datasets and use cosine similarity for ranking.

[1]We acknowledge that expecting a random permutation does not factor in the demographic or stylistic "relevance" of non-query authors to the query author. For example, an author who shares a regional dialect with the query author is more relevant to the query in this regard. While this notion of relevance is useful for attribution, it can be unfair in forensic settings as if a member of Demographic X is a suspect of a crime, undue scrutiny should not be brought to all members of Demographic X. While imperfect, a random permutation is a reasonable calibration to benchmark systems against and measure the magnitude of bias towards certain authors.

3.1 Experimental Setup

Datasets (i) Reddit: We use the evaluation partition of the dataset by Andrews and Bishop (2019) with 111,396 candidate authors and 25,000 query authors. (ii) Bloggers: We select 9000 bloggers from the authorship corpus of Blogger posts (Schler et al., 2006) and use 2500 of them as queries. (iii) Fanfiction: We randomly select 20,000 fanfiction authors, and use 7500 of them as queries.

Models We use five text embedding models: 1. SBERT (Reimers and Gurevych, 2019): A sentence transformer model based on DistilRoBERTa (Sanh et al., 2019) 2. LUAR (Rivera-Soto et al., 2021): A universal authorship embedding model trained on the Reddit Million User Dataset (Khan et al., 2021). 3. Style Embedding (Wegmann et al., 2022): A sentence transformer built on RoBERTa-base (Liu et al., 2019b) and trained for style representation. We call this model Wegmann. 4. StyleDist. (Patel et al., 2024): Another style embedding model trained on a combination of Reddit comments and synthetic data. 5. MPNet_{AR}: Microsoft's sentence transformer (MPNet_{st}) by Song et al. (2020) that we train for authorship representation. Section A.1 of the appendix provides more details on models and datasets.

We show the performance of the five embedding models on the two datasets in Table 1. In the rest of this section we look at the models' embeddings and rankings from other angles.

3.2 Misattribution Unfairness in Model Rankings

We measure MAUI_k (eq. 1) for different values of k and show the results in Table 2. Higher values show higher unfairness. Putting Table 2 and 1 together, we can see that there is no clear relationship between how good a model is in correctly attributing authors and how fair it is in misattributing authors. For example, Wegmann is the worst performing model in terms of R@8 and MRR, but it consistently shows the least unfairness. MPNet_{AR} and StyleDist. have very close MAUI scores, yet their ranking performance varies greatly. Even LUAR with very high ranking scores does not guarantee unfairness in misattribution. In fact, after SBERT, LUAR is the most unfair model to the Bloggers.

To present the unfairness from another viewpoint, we count the number of authors who carry higher risks of being ranked in top $k = 10$. Tables 3a and 3b show how these counts compare to

	Blogs		Reddit		Fanfiction	
Model	R@8	MRR	R@8	MRR	R@8	MRR
SBERT	0.61	0.48	0.15	0.10	0.28	0.22
LUAR	0.97	0.90	0.82	0.71	0.53	0.44
MPNet_{AR}	0.96	0.88	0.40	0.30	0.30	0.25
Wegmann	0.45	0.32	0.08	0.05	0.09	0.06
StyleDist.	0.68	0.55	0.09	0.06	0.16	0.12

Table 1: Recall-at-8 and Mean Reciprocal Rank scores of embedding models on three datasets.

the expected count E_{10}. The number of unfairly misattributed authors and the severity of this issue vary across models. Similar to the trends in Table 2 Wegmann is the most fair of the models on both Reddit and Blogs datasets. Again, for Blogs, LUAR is highly unfair despite its superior performance. Looking at the last column of Table 3b, LUAR's number of misattributed authors is more than MPNet_{AR}, StyleDist., and Wegmann together. Table 3 shows the counts of the number of authors subject to unfair misattribution. The differences between columns show that the extent of this misattribution risk varies for different authors. In section D of the appendix we show how the risk is distributed among authors. Looking at the authors with the most misattribution we see that for instance, with SBERT there is -at least- one author for whom the misattribution risk is almost 40x the random case.

3.3 Relation to Embedding Distribution

How the author embeddings are distributed is central to the effectiveness of attribution and fairness of misattribution in rankings. One way to analyze this relationship is via the distance of author embeddings to the centroid (center) of the embeddings.

To this end, we average all author embeddings to compute the centroid and measure distance of each author to the centroid as $1 - cosine$. We plot authors' mean rank (average of their ranking across all queries) against their distance to centroid in figure 1. For better visualization the distance values are scaled to the [0,1] range. We see strong correlations between authors' distance to the centroid and their risk of misattribution, as measured by their average rank over all queries. Across models and datasets the closer an author is to the centroid, the lower their average rank is; i.e. authors closer to the centroid are ranked higher on average. Each author is ranked for queries that are chosen at random, therefore an author's average ranking should not correlate with their distance from the centroid. We expect the plots to be close to a horizontal line. The expected average rank for every author is the middle of the ranked list i.e., half the size of haystack. Among the models we compare, Wegmann curves are closest to the ideal. Note that these correlations depend largely on how the embeddings are distributed. Indeed, we find that distance distributions vary considerably across different models. See Figure 2 in Appendix Section B.

3.4 Unfairness of missed attribution

The centroid analyses can also help us understand which authors are harder to find i.e., ones who are not recognized as the author of their own documents. To assess how often an author is ranked high for their own documents, we average their reciprocal ranks over multiple query subsets drawn from their documents[2]. Authors that are ranked close to the top when queried have higher MRRs than those who are not.

We then test three hypotheses relating MRR to authors' distance from centroid: (i) Authors with higher MRR are further away from the centroid compared to authors with lower MRR, (ii) Authors with higher MRR are further away from the centroid compared to a random subset of authors. (iii) Authors with lower MRR are closer to the center than a random subset of authors. We use the Mann-Whitney U test [3] (Mann and Whitney, 1947) to accept or reject our hypotheses. Table 7 in Appendix shows statistics for all tests. The numbers show that for Reddit the first and the second hypotheses are statistically supported for all models and the third only for LUAR. For blogs, only the third hypothesis for SBERT is rejected. See figures 3 and 4 for visualizations of this phenomenon. To summarize, while authors closer to centroid are more likely to be ranked higher when queried for

[2]Specifically, we perform four queries per each query author, each comprising four of their documents.

[3]This test does not assume normality.

	k			
Model	5	10	15	20
SBERT	0.20	0.31	0.36	0.39
LUAR	0.06	0.12	0.16	0.18
MPNet_{AR}	0.09	0.17	0.22	0.25
Wegmann	0.03	0.09	0.13	0.15
StyleDist.	0.07	0.15	0.19	0.22

(a) Reddit

	k			
Model	5	10	15	20
SBERT	0.24	0.36	0.37	0.40
LUAR	0.15	0.26	0.27	0.31
MPNet_{AR}	0.12	0.23	0.23	0.27
Wegmann	0.06	0.14	0.13	0.17
StyleDist.	0.11	0.22	0.21	0.25

(b) Blogs

	k			
Model	5	10	15	20
SBERT	0.21	0.30	0.34	0.36
LUAR	0.17	0.25	0.28	0.30
MPNet_{AR}	0.17	0.25	0.28	0.30
Wegmann	0.08	0.12	0.14	0.15
StyleDist.	0.12	0.18	0.21	0.22

(c) Fanfiction

Table 2: (MAUI_k) for different k values across models and datasets. MAUI_k is defined in section 2 as a measure of an authorship attribution system's unfairness in misattributing authors. The scores range between 0 (most fair) and 1 (most unfair).

	$> 2 \times E_{10}$	$> 4 \times E_{10}$	$> 5 \times E_{10}$
SBERT	8487	2582	1599
LUAR	4290	242	54
MPNet_{AR}	6054	701	299
Wegmann	2967	18	3
StyleDist.	5411	382	106

(a) Reddit

	$> 2 \times E_{10}$	$> 4 \times E_{10}$	$> 5 \times E_{10}$
SBERT	858	296	214
LUAR	821	189	96
MPNet_{AR}	816	122	56
Wegmann	496	9	1
StyleDist.	789	104	33

(b) Blogs

	$> 2 \times E_{10}$	$> 4 \times E_{10}$	$> 5 \times E_{10}$
SBERT	1319	328	104
LUAR	1161	170	38
MPNet_{AR}	1155	181	43
Wegmann	316	0	0
StyleDist.	780	32	0

(c) Fanfiction

Table 3: Number of authors who are ranked in top 10 more than the expected number i.e. E_{10}. Note that the size of the haystack and the number of the queries are different across datasets, hence the huge difference in the numbers between the three tables.

Model	**Reddit**	**Blogs**	**Fanfic**
LUAR	9.75	10.0	12.0
SBERT	39.00	21.75	5.8
Wegmann	4.50	4.25	3.6
MPNet	12.25	10.25	12.0
StyleDist	8.25	7.0	19.6

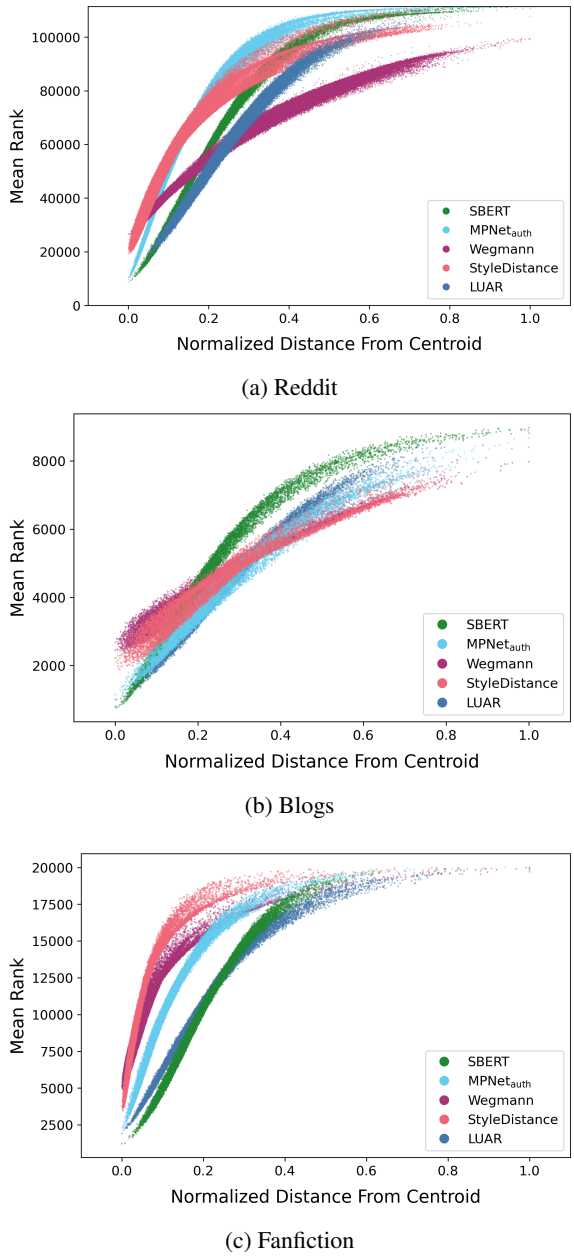
Table 4: Extreme cases of unfair misattribution risk. Numbers show the ratio of the number of times an author is ranked in top 10 to E_{10} for the author bearing the highest risk of misattribution.

other authors' documents, they are not necessarily ranked high for their own. These results also demonstrate the importance of embedding distributions to the fairness of authorship rankings.

4 Related Work

Automatic authorship analysis has a rich body of work (see Huang et al., 2025; Tyo et al., 2022), with a heavy emphasis on usefulness and reliability of features used in authorship analysis (e.g. Chaski (2001); Baayen et al. (2002), or the effects of the attribution setup (e.g. Stamatatos (2013); Sari et al. (2018)). While these provide general insights into authorship attribution, to the best of our knowledge, there have been no specific studies that focus on notions of fairness in authorship attribution. Fairness of NLP models has been studied in many application domains including dialogue (Liu et al., 2019a), language modeling (Cao et al., 2022; Qian et al., 2022), text generation (Fleisig et al., 2023), classification (Pruksachatkun et al., 2021) and clinical NLP (Meng et al., 2022). Gallegos et al. (2024) survey research on bias and fairness in large language models focusing on metrics, probing datasets, and bias mitigation techniques. There have also been extensive research on fairness and bias related metrics. Czarnowska et al. (2021) survey, categorize, and compare some of the fairness and bias related metrics.

The closest to our work is research on fairness of embedding-based document retrieval systems. Their main concerns are: group fairness (Yang



(a) Reddit

(b) Blogs

(c) Fanfiction

Figure 1: Relationship between authors' average rank and their distance from the centroid. Distances are min-max normalized. Authors' mean rank is highly correlated with their distance from centroid.

and Stoyanovich, 2016), individual fairness (Biega et al., 2018; García-Soriano and Bonchi, 2021), their trade-off with each other and with relevance (Gao and Shah, 2019), and recently unfairness in ranking with LLMs (Wang et al., 2024). These works aim to ensure relevant documents from certain individuals or groups are fairly represented in the top ranks. In contrast, in our authorship setting, the focus is on the risks of being included in the top ranks in forensic or law enforcement settings.

5 Conclusion

Authorship attribution carries profound risks in settings such as forensic and law enforcement uses. In these settings, measures of effectiveness alone are not adequate for evaluating and understanding the impact of misattribution. To this end, we argued that we also need to understand if the rankings induced by attribution models distribute misattribution risks equally. The unfairness measure we introduced helps quantify these risks. Our empirical measurements of models used to produce authorship embeddings shows that in an embed-and-rank approach there are many authors who are at a substantially higher risk of being ranked in the top k. We further showed that this risk correlates with how the embeddings are distributed. Our findings call for careful consideration of such notions of misattribution fairness in evaluation, development, and deployments of authorship analysis systems.

Limitations

In practice, the anticipated "fair" ranking of the haystack authors is not entirely random, but rather a function of authors' relatedness to the query. The distribution of queries therefore impacts the "most fair" baseline, and consequently the unfairness measurements. This impact is not accounted for in this paper. Our results hold for the set of queries selected for in our experiments, which were chosen at random. It is possible different selection methods of queries may impact unfairness measurements. Additionally, this work's focus is limited to cases where over-attribution is undesired. A broader study would also cover cases where under-attribution is problematic.

Ethical Considerations

Potential Misuse Results and analyses presented in this paper are meant to prompt researchers in authorship attribution to focus on unfairness, assessing it, and preventing its potential harms. People and institutions using authorship attribution may misuse these results to justify automatically preempting groups of candidate authors. We do not investigate characteristics of authors who are easy or difficult to track down. Nevertheless, a malicious agent may reproduce our experiments, analyze hard-to-find authors (near-centroid authors), and use characteristics of their writing as an authorship obfuscation method.

Our Work We report licenses and sources of every artifact we used in Appendix A. We did not investigate individuals or use demographic information (e.g. reported age and gender published in the Blogs corpus) about authors in any of our experiments and operate under the assumption that no author should have significantly greater than random expectation of appearing in the top k results when they are not the query author of interest regardless of demographics.

Acknowledgments

This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract #2022-22072200005. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Nicholas Andrews and Marcus Bishop. 2019. Learning invariant representations of social media users. *Preprint*, arXiv:1910.04979.

Harald Baayen, Hans Van Halteren, Anneke Neijt, and Fiona Tweedie. 2002. An experiment in authorship attribution. In *6th JADT*, volume 1, pages 69–75. Citeseer.

Asia J. Biega, Fernando Diaz, Michael D. Ekstrand, and Sebastian Kohlmeier. 2020. Overview of the trec 2019 fair ranking track. *Preprint*, arXiv:2003.11650.

Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*.

Steven Burrows, Alexandra L. Uitdenbogerd, and Andrew Turpin. 2009. Application of information retrieval techniques for source code authorship attribution. In *Database Systems for Advanced Applications*, pages 699–713, Berlin, Heidelberg. Springer Berlin Heidelberg.

Yang Trista Cao, Yada Pruksachatkun, Kai-Wei Chang, Rahul Gupta, Varun Kumar, J. Dhamala, and A. G. Galstyan. 2022. On the intrinsic and extrinsic fairness evaluation metrics for contextualized language representations. *ArXiv*, abs/2203.13928.

Carole E Chaski. 2001. Empirical evaluations of language-based author identification techniques. *Forensic linguistics*, 8:1–65.

Paula Czarnowska, Yogarshi Vyas, and Kashif Shah. 2021. Quantifying social biases in NLP: A generalization and empirical comparison of extrinsic fairness metrics. *Transactions of the Association for Computational Linguistics*, 9:1249–1267.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Eve Fleisig, Aubrie Amstutz, Chad Atalla, Su Lin Blodgett, Hal Daumé, Alexandra Olteanu, Emily Sheng, Dan Vann, and Hanna M. Wallach. 2023. Fairprism: Evaluating fairness-related harms in text generation. In *Annual Meeting of the Association for Computational Linguistics*.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179.

Ruoyuan Gao and Chirag Shah. 2019. How fair can we go: Detecting the boundaries of fairness optimization in information retrieval. *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*.

David García-Soriano and Francesco Bonchi. 2021. Maxmin-fair ranking: Individual fairness under group-fairness constraints. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

Baixiang Huang, Canyu Chen, and Kai Shu. 2025. Authorship attribution in the era of llms: Problems, methodologies, and challenges. *Preprint*, arXiv:2408.08946.

Mike Kestemont, Enrique Manjavacas, Ilia Markov, Janek Bevendorff, Matti Wiegmann, Efstathios Stamatatos, Martin Potthast, and Benno Stein. 2020. Overview of the cross-domain authorship verification task at pan 2020. In *Conference and Labs of the Evaluation Forum*.

Mike Kestemont, Efstathios Stamatatos, Enrique Manjavacas, Walter Daelemans, Martin Potthast, and Benno Stein. 2019. Overview of the cross-domain authorship attribution task at pan 2019. In *Conference and Labs of the Evaluation Forum*.

Aleem Khan, Elizabeth Fleming, Noah Schofield, Marcus Bishop, and Nicholas Andrews. 2021. A deep metric learning approach to account linking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5275–5287, Online. Association for Computational Linguistics.

Haochen Liu, Jamell Dacon, Wenqi Fan, Hui Liu, Zitao Liu, and Jiliang Tang. 2019a. Does gender matter? towards fairness in dialogue systems. *ArXiv*, abs/1910.10486.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Coulthard M., Johnson A., and Wright D. 2016. *An Introduction to Forensic Linguistics: Language in Evidence (2nd ed.)*. Routledge.

Henry B. Mann and Douglas R. Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18:50–60.

Chuizheng Meng, Loc Trinh, Nan Xu, James Enouen, and Yan Liu. 2022. Interpretability and fairness evaluation of deep learning models on mimic-iv dataset. *Scientific Reports*, 12.

Ajay Patel, Jiacheng Zhu, Justin Qiu, Zachary Horvitz, Marianna Apidianaki, Kathleen McKeown, and Chris Callison-Burch. 2024. Styledistance: Stronger content-independent style embeddings with synthetic parallel examples. *Preprint*, arXiv:2410.12757.

Yada Pruksachatkun, Satyapriya Krishna, J. Dhamala, Rahul Gupta, and Kai Wei Chang. 2021. Does robustness improve fairness? approaching fairness with word substitution robustness methods for text classification. *ArXiv*, abs/2106.10826.

Rebecca Qian, Candace Ross, Jude Fernandes, Eric Michael Smith, Douwe Kiela, and Adina Williams. 2022. Perturbation augmentation for fairer nlp. In *Conference on Empirical Methods in Natural Language Processing*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

Rafael A. Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y. Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. 2021. Learning universal authorship representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 913–919, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Yunita Sari, Mark Stevenson, and Andreas Vlachos. 2018. Topic or style? exploring the most useful features for authorship attribution. In *International Conference on Computational Linguistics*.

Jonathan Schler, Moshe Koppel, Shlomo Engelson Argamon, and James W. Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Preprint*, arXiv:2004.09297.

Efstathios Stamatatos. 2013. On the robustness of authorship attribution based on character n -gram features. *Journal of law and policy*, 21:7.

Urszula Stańczyk and Krzysztof A Cyran. 2007. Machine learning approach to authorship attribution of literary texts. *International journal of applied mathematics and informatics*, 1(4):151–158.

Michael Tschuggnall, Benjamin Murauer, and Günther Specht. 2019. Reduce & attribute: Two-step authorship attribution for large-scale problems. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 951–960, Hong Kong, China. Association for Computational Linguistics.

Jacob Tyo, Bhuwan Dhingra, and Zachary C. Lipton. 2022. On the state of the art in authorship attribution and authorship verification. *Preprint*, arXiv:2209.06869.

Andrew Wang, Cristina Aggazzotti, Rebecca Kotula, Rafael A. Rivera Soto, Marcus Bishop, and Nicholas Andrews. 2023. Can authorship representation learning capture stylistic features? *Transactions of the Association for Computational Linguistics*, 11:1416–1431.

Yuan Wang, Xuyang Wu, Hsin-Tai Wu, Zhiqiang Tao, and Yi Fang. 2024. Do large language models rank fairly? an empirical study on the fairness of LLMs as rankers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5712–5724, Mexico City, Mexico. Association for Computational Linguistics.

Anna Wegmann, Marijn Schraagen, and Dong Nguyen. 2022. Same author or just same topic? towards content-independent style representations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 249–268, Dublin, Ireland. Association for Computational Linguistics.

Zichen Wen, Dadi Guo, and Huishuai Zhang. 2024. Aidbench: A benchmark for evaluating the authorship identification capability of large language models. *ArXiv*, abs/2411.13226.

Ke Yang and Julia Stoyanovich. 2016. Measuring fairness in ranked outputs. *Preprint*, arXiv:1610.08559.

Ying Zhao and Justin Zobel. 2007. Searching with style: Authorship attribution in classic literature. In *ACM international conference proceeding series*, volume 244, pages 59–68. Citeseer.

A Technical Details of Experiments

A.1 Models and Datasets

Blogs Data

We use the corpus of Blogger posts collected by Schler et al. (2006) as it is commonly used in authorship analysis research. This dataset "may be freely used for non-commercial research purposes."[4] We filter for authors with more than 10 and fewer than 200 posts. From those we pick 9000 candidate authors, 2500 of which are used as queries and needles. Each author in the haystack has 16 blog posts (randomly selected) and each author in the queries has 10.

Reddit Data

For evaluation we use the evaluation partition of the Reddit dataset created by Andrews and Bishop (2019). It has 111,396 candidate authors and 25,000 query authors. Each author in the haystack has 16 Reddit comments, and each author in the queries has 16 Reddit comments as well. For finetuning MPNet_{AR} we use a part of the Million User Dataset by Khan et al. (2021).

Fanfiction Data

We use a subset of the fanfiction dataset from PAN19 (Kestemont et al., 2019) and PAN20 (Kestemont et al., 2020) for evaluation. This subset consists of 20000 haystack authors, 7500 of whom are used as queries.

A.2 Finetuning MPNet_{st}

We choose to finetune MPNet_{st} (Song et al., 2020) because it has a different structure than the BERT-based models. Also, all-mpnet-base-v2 [5] is a top-performing sentence transformer. Out of 12 layers we keep 8 frozen for training. This model is licensed under the apache-2.0 license. We use cached multiple-negative ranking loss implemented in the sentence-transformers library[6]. Training with multiple negatives has shown to be very effective in the case of LUAR (Rivera-Soto et al., 2021). We set the learning rate to $5e^{-5}$ with a linear scheduler. Maximum sequence length is set to 512, and we train with a batch size of 200. We train for 5000 steps on the training subset of Reddit data by Khan et al. (2021). Training computation was done on an NVIDIA RTX A6000 GPU.

[4] See https://u.cs.biu.ac.il/ koppel/BlogCorpus.htm

[5] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[6] https://www.sbert.net/index.html

Model	Base Sentence-Transformer	Training Objective	Training Data	# of Parameters	License
SBERT	DistilRoBERTa	-	-	82M	apache-2.0
LUAR	paraphrase-distilroberta-base-v1	Batch Contrastive Loss*	Reddit (MUD)	82M	apache-2.0
MPNet_{AR}	all-mpnet-base-v2	Multiple-Negative Ranking Loss	Reddit(MUD)	109M	apache-2.0
Wegmann	RoBERTa-base	Triplet Loss	Reddit	124M	mit
StyleDist.	RoBERTa-base	Contrastive Loss	Reddit + Synthetic Data	124M	mit

Table 5: Overview of embedding models we use in experiments. We perform inference on these models using an NVIDIA TITAN Xp GPU.

A.3 Indexing and Retrieving Embeddings

We use Faiss (Douze et al., 2024)[7] for efficient indexing of author embeddings and performing fast searches. All the indexing and search computations are done on CPU. Faiss library is under an MIT-license.

A.4 Libraries and Packages

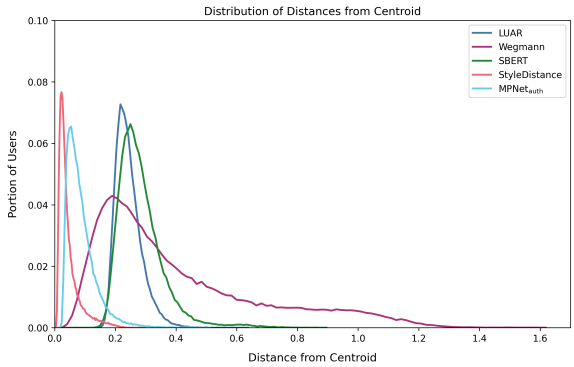
Here is a list of major python libraries we use (with no particular order):

- torch==2.3.1
- sentence-transformers==3.3.1
- transformers==4.45.2
- tensorflow-datasets==4.9.6
- scikit-learn==1.5.0
- scipy==1.14.0
- numpy==1.26.4
- nltk==3.8.1
- matplotlib==3.9.2
- seaborn==0.13.2
- faiss-cpu==1.8.0.post1

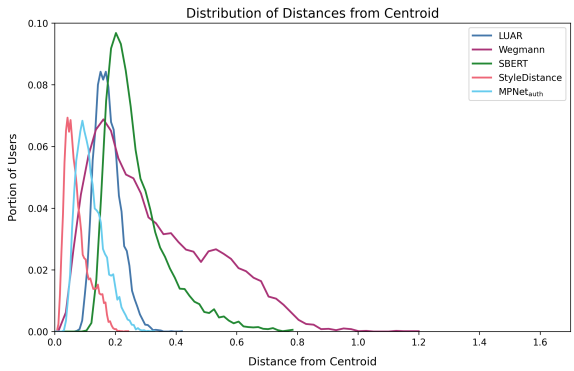
B Embedding Distributions

Since authorship attribution in this setting is based entirely upon embeddings and their distances, we use a simple distance-to-centroid measure to characterize the distribution of author embeddings under different models. Intuitively, if all author embeddings are clustered close to each other then the distances to the centroid will be small. If they are spread out then the distances to the centroid will be large. Per each dataset and model we find the centroid of all haystack author embeddings, then calculate each author's distance from this centroid. The distance measure is $1 - cosine$, so it can take values from 0 to 2.

[7]https://github.com/facebookresearch/faiss



(a) Reddit



(b) Blogs

Figure 2: Distribution of authors' distinctness (i.e. their distance from the centroid computed as: $1 - cosine$)

Figure 2 shows the distribution of the distance value per each model for Reddit and Blogs datasets. The diversity of the distributions in terms of domain and curve shape inspired us to investigate how it affects the effectiveness and fairness of authorship attribution.

Model	Max	Mean	Std
LUAR	9.75	1.62	0.50
SBERT	39.00	2.37	1.67
Wegmann	4.50	1.50	0.32
MPNet	12.25	1.79	1.74
StyleDist	8.25	1.69	0.56

(a) Reddit

Model	Max	Mean	Std
LUAR	10.0	2.07	1.10
SBERT	21.75	2.61	2.12
Wegmann	4.25	1.58	0.40
MPNet	10.25	1.92	0.85
StyleDist	7.00	1.84	0.70

(b) Blogs

Model	Max	Mean	Std
LUAR	12.0	2.03	1.05
SBERT	19.6	2.32	1.58
Wegmann	3.6	1.54	0.39
MPNet	12.0	2.04	1.09
StyleDist	5.80	1.75	0.64

(c) Fanfiction

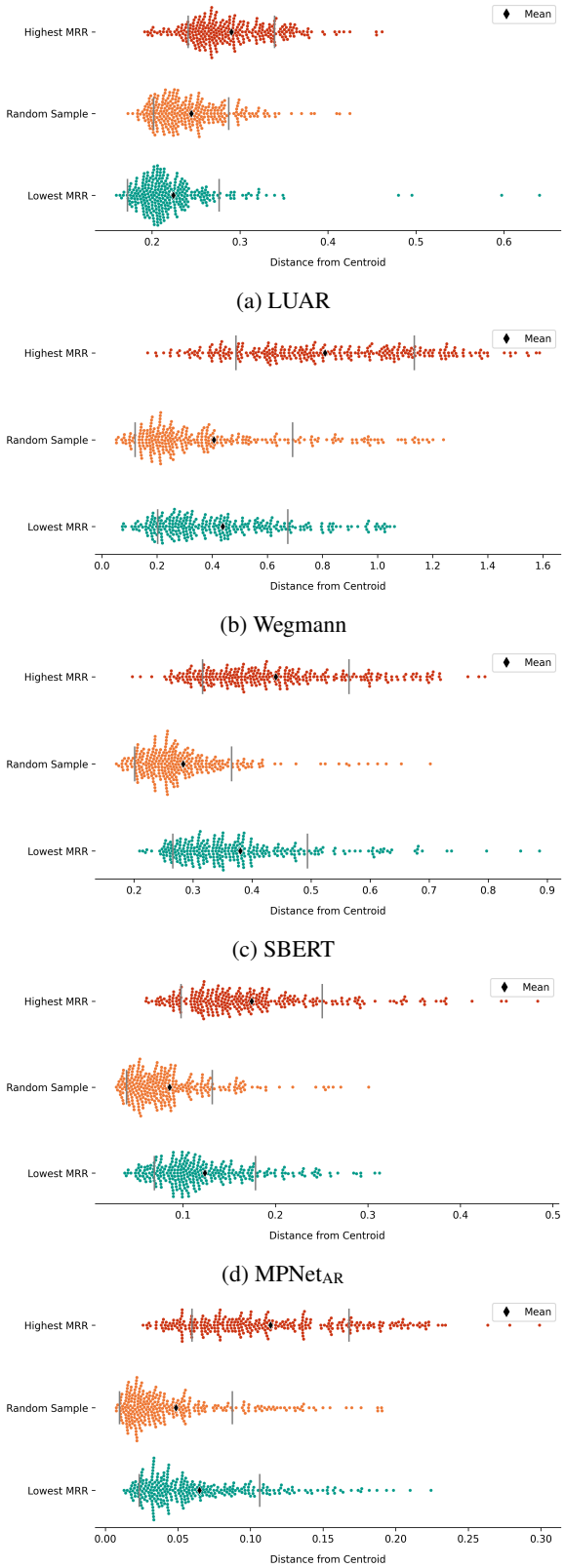
Table 6: Variation of the risk of unfair misattribution across different authors measured by the ratio of the number of times an author is ranked in top 10 to E_{10}.

C Analysis of Needle Authors' Distance to Centroid

Test statistics corresponding to section 3.4 are presented in 7. Additionally, we sample 300 Reddit authors with highest MRRs, 300 authors with lowest MRRs, and 300 random authors to visualize the relationship between needle authors' MRR and their distance from the centroid. Figure 3 shows the distribution of the sampled authors' distances from the centroid. Similar plots for Blogs experiments are in figure 4.

D Risk of Misattribution for Different Authors

Not all authors who are subject to over-misattribution carry the same risk. Per dataset and model, we obtain the ratio of the number of times an author is ranked in top 10 to E_{10}: $u_j^{10} = \frac{c_j^{10}}{E_{10}}$. For all authors contributing to the MAUI_{10} score (eq. 1) this ratio is more than 1. Table 3 shows for how many authors this ratio is higher than 2, 4, and 5. Table 6 shows the maximum, mean, and standard deviation of u^{10} across authors.



(a) LUAR

(b) Wegmann

(c) SBERT

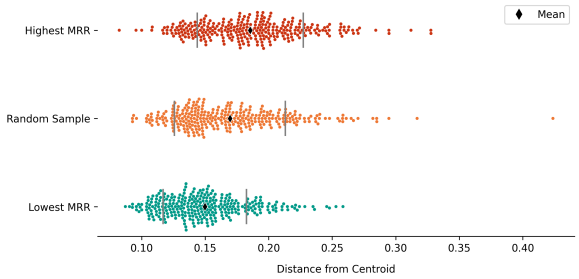
(d) MPNet_{AR}

(e) StyleDist.

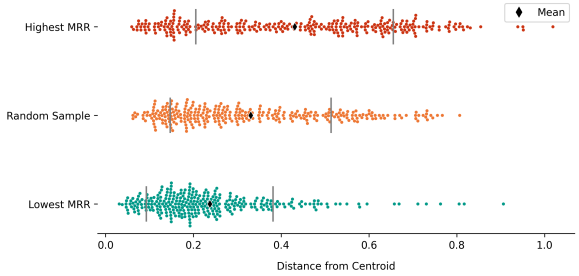
Figure 3: Reddit - Distribution of Needle Authors' Distances from the Centroid. A comparison between random samples of authors, a group of easily found authors and a group of hard to find authors.

		Reddit			Blogs		
		Hyp. (i)	Hyp. (ii)	Hyp. (iii)	Hyp. (i)	Hyp. (ii)	Hyp. (iii)
LUAR	U	79199.0	71937.5	30635.0	67607.0	56918.5	34128.0
	p	$1.12e^{-58}$	$3.46e^{-37}$	$6.63e^{-12}$	$8.91e^{-27}$	$9.91e^{-9}$	$1.52e^{-7}$
Wegmann	U	74052.0	75528.0	50676.0	67255.0	55346.5	31362.0
	p-value	$6.36e^{-43}$	$3.51e^{-47}$	1.0	$5.22e^{-6}$	$5.49e^{-7}$	$6.66e^{-11}$
SBERT	U	59269.0	78719.5	70489.0	65060.0	60090.5	41619.0
	p-value	$9.05e^{-12}$	$4.22e^{-57}$	1.0	$1.72e^{-21}$	$5.9e^{-13}$	0.055
MPNet_{AR}	U	64956.0	79814.5	66651.5	74715.0	61172.5	29477.0
	p-value	$2.75e^{-21}$	$9.91e^{-61}$	1.0	$8.26e^{-45}$	$1.30e^{-14}$	$1.33e^{-13}$
StyleDist.	U	70389.0	79633.0	62192.5	76030.0	63759.5	30604.5
	p-value	$2.94e^{-33}$	$4.03e^{-60}$	1.0	$1.12e^{-48}$	$4.97e^{-19}$	$6e^{-12}$

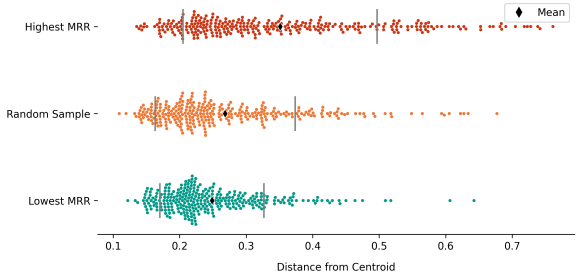
Table 7: Mann-Whitney U-statistics and p-values of testing three hypotheses about our authorship attribution experiments: (i) Authors with higher MRR are further away from the centroid compared to authors with lower MRR, (ii) Authors with higher MRR are further away from the centroid compared to a random subset of authors. (iii) Authors with lower MRR are closer to the center than a random subset of authors.



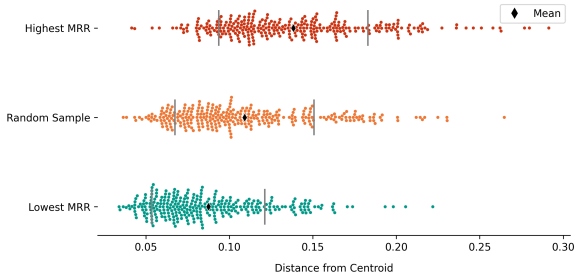
(a) LUAR



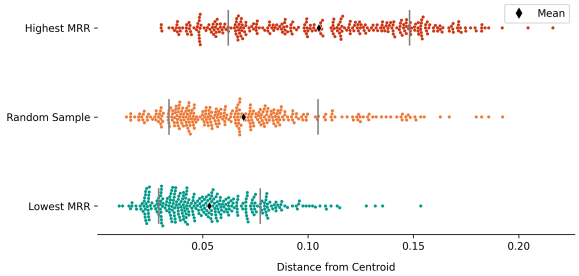
(b) Wegmann



(c) SBERT



(d) MPNet_{AR}



(e) MPNet_{AR}

Figure 4: Blogs - Distribution of Needle Authors' Distances from the Centroid. A comparison between random samples of authors, a group of easily found authors and a group of hard to find authors.

Zero-Shot Text-to-Speech for Vietnamese

Thi Vu, Linh The Nguyen, Dat Quoc Nguyen
Movian AI, Vietnam
{thivuxy, toank45sphn, datquocnguyen}@gmail.com

Abstract

This paper introduces PhoAudiobook, a newly curated dataset comprising 941 hours of high-quality audio for Vietnamese text-to-speech. Using PhoAudiobook, we conduct experiments on three leading zero-shot TTS models: VALL-E, VoiceCraft, and XTTS-V2. Our findings demonstrate that PhoAudiobook consistently enhances model performance across various metrics. Moreover, VALL-E and VoiceCraft exhibit superior performance in synthesizing short sentences, highlighting their robustness in handling diverse linguistic contexts. We publicly release PhoAudiobook to facilitate further research and development in Vietnamese text-to-speech.

1 Introduction

Text-to-speech (TTS) synthesis has witnessed significant advancements in recent years. State-of-the-art TTS systems typically use a cascaded pipeline that consists of an acoustic model and a vocoder, with mel-spectrograms serving as intermediate representations (Ren et al., 2019; Li et al., 2019; Tan et al., 2024). These advanced TTS systems can synthesize high-quality speech for single or multiple speakers (Kim et al., 2021; Liu et al., 2022).

Zero-shot TTS has emerged as a promising approach to overcome the limitations of traditional TTS systems in generalizing to unseen speakers. By leveraging techniques such as speaker adaptation and speaker encoding, zero-shot TTS aims to synthesize speech for new speakers using only a few seconds of reference audio (Arik et al., 2018; Wang et al., 2020; Cooper et al., 2020; Wu et al., 2022; Casanova et al., 2022, 2024). Recent works have explored the application of language modeling approaches to zero-shot TTS, achieving impressive results. For example, VALL-E (Wang et al., 2023) introduces a text-conditioned language model trained on discrete audio codec tokens, enabling TTS to be treated as a conditional codec language modeling task. VoiceCraft (Peng et al., 2024) casts both sequence infilling-based speech editing and continuation-based zero-shot TTS as a left-to-right language modeling problem by rearranging audio codec tokens.

Despite advancements in zero-shot TTS, its application to low-resource languages remains challenging. These languages often lack the large-scale, high-quality datasets needed to train robust TTS models (Gutkin et al., 2016; Chen et al., 2019; Lux et al., 2022; Ngoc et al., 2023; Huang et al., 2024). Also, linguistic and phonetic differences between languages introduce additional challenges in adapting existing models to new languages. As a result, the performance of zero-shot TTS systems in low-resource languages is often limited, hindering their practical usability.

In this paper, we focus on advancing zero-shot TTS for Vietnamese. Our contributions are:

- **We present PhoAudiobook**, a 941-hour high-quality long-form speech dataset that overcomes the limitations of existing Vietnamese datasets, which usually contain audio samples shorter than 10 seconds. The pipeline to create this dataset can be easily adapted to other languages.
- **We conduct a comprehensive experimental study** to evaluate the performance of three state-of-the-art zero-shot TTS models: VALL-E, VoiceCraft, and XTTS-v2 (Casanova et al., 2024). Using a combination of objective and subjective metrics across multiple benchmark datasets, our results demonstrate that XTTS-v2 trained on PhoAudiobook outperforms its counterpart trained on an existing dataset. Additionally, VALL-E and VoiceCraft exhibit robustness in synthesizing varied input lengths.
- **We publicly release PhoAudiobook** at https://huggingface.co/datasets/thivux/phoaudiobook for non-commercial purposes.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1042–1049
July 27 - August 1, 2025

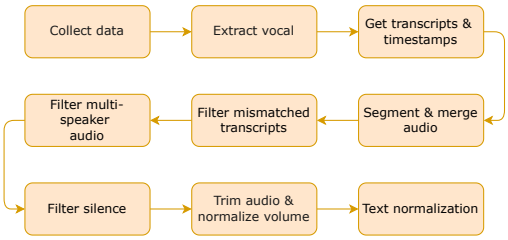


Figure 1: PhoAudiobook creation pipeline.

2 Dataset

2.1 PhoAudiobook

Figure 1 illustrates the creation process of our PhoAudiobook dataset.

First, we collect raw Vietnamese audiobook data from the publicly accessible website `https://sachnoiviet.net`. This raw dataset includes 23K hours of content from 2,697 audiobooks, narrated by 735 distinct speakers. Next, we use `demucs` to extract the vocal track, effectively removing any background music or sound effects (Défossez, 2021; Rouard et al., 2023). We then employ the multilingual `Whisper-large-v3` model to generate transcriptions and corresponding timestamps for the audio data (Radford et al., 2023). The output from `Whisper-large-v3` includes transcripts for short audio segments, usually a few seconds long, along with their corresponding timestamps. These segments are often aligned with natural pauses in speech. We then concatenate *successive* audio segments and their corresponding transcripts to create longer audio samples and transcriptions, each lasting between 10 and 20 seconds. To ensure the quality of the transcriptions, we process these merged samples using the state-of-the-art Vietnamese ASR model, `PhoWhisper-large` (Le et al., 2024). We then retain only the samples where the `Whisper-large-v3`-based transcription matches exactly with the transcription output from `PhoWhisper-large`. Furthermore, we tackle the challenge of multi-speaker audio. We use the `wav2vec2-bartpho` model to identify and filter out short audio samples containing multiple speakers, ensuring that all audio segments associated with a particular speaker are indeed spoken by that individual.[1]

To reduce excessive silence in the audio data, we exclude samples with transcripts shorter than 25 words and trim silence from the beginning and end of each sample. Additionally, we use the `sox` library to normalize audio volume levels for maintaining consistency and avoiding abrupt loudness throughout the dataset.[2] Finally, we standardize the transcriptions through a text normalization step, which includes converting text to lowercase, adding appropriate punctuation, and normalizing numerical expressions into their text form (e.g., "43" becomes "forty three"). We carry out this text normalization step using a sequence-to-sequence model, which we develop by fine-tuning the pre-trained `mbart-large-50` model (Liu et al., 2020) on a Vietnamese dataset consisting of unnormalized input and normalized output text pairs.[3]

The data creation process described above results in a refined 1,400-hour audio corpus. To ensure balanced speaker representation, we limit each speaker to a maximum of 4 hours of audio. This results in a high-quality dataset comprising **941** hours of audio from **735** speakers. From the remaining $1,400 - 941 = 559$ hours of audio, we sample 0.8 hour of audio from 20 speakers to construct a "seen" speaker test set. Additionally, we split the 941 hours of audio from 735 speakers into three sets (on speaker level): a training set containing 940 hours from 710 speakers, a validation set with 0.5 hours from 5 speakers, and an "unseen" speaker test set comprising 0.4 hours from 20 speakers who have the shortest total audio durations. Here, the 20 speakers in the "seen" speaker test set are part of the 710 speakers used for training.

We conduct a post-processing step to manually inspect each audio sample and its corresponding transcription from both the "seen" and "unseen" speaker test sets. This process results in all correct transcriptions in the test sets of PhoAudiobook.

2.2 Dataset analysis

Table 1 presents the characteristics of our dataset – PhoAudiobook, in comparison to other Vietnamese speech datasets, including VinBigData (VinBigData, 2023), VietnamCeleb (Pham et al., 2023), the VLSP 2020 ASR Challenge,[4] BUD500 (Pham et al., 2024), and viVoice (Gia et al., 2024).

Duration: PhoAudiobook, with 941 hours, is the second-largest dataset, closely following viVoice,

[1] `https://huggingface.co/nguyenvulebinh/wav2vec2-bartpho`

[2] `https://sourceforge.net/projects/sox`

[3] `https://huggingface.co/datasets/nguyenvulebinh/spoken_norm_pattern`

[4] `https://vlsp.org.vn/vlsp2020/eval/asr`

Dataset	Duration (h)	Mean Dur. (s)	25% Dur. (s)	75% Dur. (s)	Domain	SI-SNR (dB)	# Speakers	Rate (wpm)	Fs (Hz)
VinBigData	101	6.47	3.54	8.09	General-purpose	4.77	Unknown	229	16000
VietnamCeleb	187	7.74	2.84	9.64	Unknown	3.89	Unknown	No transcripts	16000
VLSP	243	4.37	2.43	5.21	Unknown	4.35	Unknown	242	16000
BUD500	462	2.56	2.11	2.94	General-purpose	4.22	Unknown	224	16000
viVoice	1016	4.12	1.96	5.55	General-purpose	4.81	Unknown	243	24000
PhoAudiobook	941	11.66	10.63	12.18	Audiobooks	4.91	735	201	16000

Table 1: Characteristics of PhoAudiobook and other speech datasets for Vietnamese.

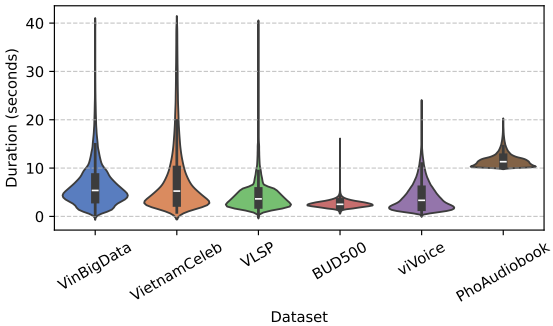


Figure 2: Duration distributions of datasets. Audio samples are capped at 40 seconds for visualization purposes.

which has 1,016 hours. The other datasets are considerably smaller, ranging from 101 to 462 hours. Figure 2 shows that previous datasets primarily consist of audio segments shorter than 10 seconds. PhoAudiobook addresses this limitation by providing audio samples ranging from 10 to 20 seconds.

Domain: PhoAudiobook is derived from audiobooks, typically recorded with professional equipment in controlled environments, ensuring high-quality audio. In contrast, other datasets are general-purpose (e.g., news, YouTube videos, conversations) and may include audio recorded on consumer devices in uncontrolled settings, often with background noise. However, general-purpose datasets have the advantage of covering diverse topics and speaking styles.

Signal-to-Noise Ratio (SI-SNR): Using an SI-SNR estimator from the speechbrain toolkit (Ravanelli et al., 2024), we calculated SI-SNR across 1000 randomly sampled audio from each dataset. PhoAudiobook achieves the highest SI-SNR, surpassing all other datasets, including viVoice.

Speaker Information: PhoAudiobook is the only dataset with explicit speaker identity, and therefore, number of speakers (735).

Speaking Rate (wpm): Among the four datasets with transcripts, PhoAudiobook has the lowest speaking rate of words per minute. This reflects the nature of the dataset, which features long-form audio where speakers naturally pause and rest.

Sampling Rate: All datasets except viVoice use a standard sampling rate of 16000 Hz, which is a widely used sampling rate for speech data.

PhoAudiobook is comparable in total duration to viVoice, however, it offers several advantages:

- **Text Normalization:** viVoice lacks text normalization, which limits its suitability for certain TTS models. In contrast, PhoAudiobook offers normalized transcripts, enhancing compatibility with these models.
- **Audio Quality:** The unnormalized audio waveforms in viVoice may cause quality issues like distortion and inconsistent volume. In contrast, PhoAudiobook ensures audio waveforms are normalized for consistent quality.
- **Speaker ID:** viVoice does not provide speaker IDs for individual audio samples, but uses YouTube channel names as a proxy. This approach can be problematic when a YouTube channel features multiple speakers, limiting the use of this dataset to models that do not require speaker identification. In contrast, PhoAudiobook provides distinct speaker IDs for each audio sample, ensuring its broader applicability for speaker-dependent tasks.

3 Empirical approach

3.1 Models & Training data augmentation

We conduct experiments using 3 state-of-the-art zero-shot TTS models: VALL-E (Wang et al., 2023), VoiceCraft (Peng et al., 2024), and XTTS-v2 (Casanova et al., 2024). (i) VALL-E, a pioneering language model-based approach, treats text-

to-speech (TTS) as a conditional language modeling task. It utilizes discrete acoustic tokens derived from a neural audio codec and leverages massive datasets to achieve impressive zero-shot, in-context learning capabilities. (ii) VoiceCraft, a token-infilling neural codec language model, excels in both speech editing and zero-shot TTS. It employs a Transformer decoder architecture with a novel token rearrangement procedure to generate high-quality speech. (iii) XTTS-v2 builds upon the Tortoise model (Betker, 2023), incorporating modifications for multilingual training and enhanced voice cloning. It excels in synthesizing speech for numerous languages, including low-resource ones.

To enhance data distribution and ensure our TTS models effectively handle shorter input text, we augment the PhoAudiobook training set with shorter audio clips. Specifically, we treat the PhoAudiobook training set, which consists of 940 hours, as a new raw dataset and apply our dataset creation process as detailed in Section 2.1. However, we omit (i) the step of merging short segments into longer ones and (ii) the step of excluding short samples. This augmentation phase results in an additional 554 hours of short audio, bringing the total to 940 + 554 = 1494 hours of audio for training. See implementation details on how we train VALL-E, VoiceCraft, and XTTS-v2 on this 1494-hour training set in Appendix A.

3.2 Evaluation setup

Baseline: The baseline model is viXTTS (Gia et al., 2024),[5] which is fine-tuned from the pretrained XTTS-v2 on the viVoice dataset.

Test sets: In addition to using our PhoAudiobook "seen" and "unseen" speaker test sets, we also compare our models with viXTTS on the VIVOS test set (Luong and Vu, 2016), which contains 0.75 hours of short audio data from 19 speakers. Furthermore, we randomly select 8 speakers, totaling 0.5 hours of audio, from the viVoice dataset for testing. It is important to note that viVoice is available only as a single training dataset, without a predefined training/validation/test split. Consequently, this 0.5-hour viVoice audio set is in fact used for training the baseline viXTTS.

Metrics: To compare our models and the baseline, we use objective metrics including Word Error Rate (WER), Mel-Cepstral Distortion (MCD) and F0 Root Mean Square Error (RMSE_{F0}), as well as subjective metrics Mean Opinion Score (MOS) and Similarity MOS (SMOS).

Objective metrics provide quantifiable measures of specific aspects of synthesized speech:

- **Word Error Rate (WER):** This metric assesses the intelligibility of synthesized speech by calculating the edit distance between the transcription of the synthesized speech and the ground truth transcription. Specifically, it counts the number of insertions, deletions, and substitutions needed to turn one into the other. A lower WER indicates higher intelligibility. Here, we employ the ASR model `PhoWhisper-large` (Le et al., 2024) to generate the transcription of the synthesized speech.
- **Mel-Cepstral Distortion (MCD):** This metric quantifies the spectral difference between synthesized speech and the ground truth audio. A lower MCD value indicates higher spectral similarity and better quality. We use the `pymcd`[6] package to compute the MCD.
- **F0 Root Mean Square Error (RMSE_{F0}):** This metric measures the difference in fundamental frequency (F0) between synthesized speech and the ground truth audio. A lower RMSE_{F0} suggests a better matching of intonation and prosody. We use the `Amphion` (Li et al., 2025) toolkit to compute this value.

Subjective Metrics rely on human judgments to evaluate the overall quality and naturalness of synthesized speech.

- **Mean Opinion Score (MOS):** This metric assesses the overall quality of synthesized speech, taking into account factors such as naturalness, clarity, and listening effort. Human listeners rate the speech on a scale from 1 (very poor) to 5 (excellent).
- **Similarity MOS (SMOS):** This metric evaluates the perceived speaker similarity between the speech prompt and the generated speech. Listeners rate the similarity on a scale from 1 (completely different) to 5 (identical).

To conduct the subjective evaluation, we first randomly sample one audio file from each speaker in the test set. We then hire 10 native speakers

[5] https://huggingface.co/capleaf/viXTTS

[6] https://pypi.org/project/pymcd/

	Model	PAB-S	PAB-U	VIVOS	viVoice
WER ↓	Original	0.88	0.83	5.14	4.97
	VALL-E$_{\text{PAB}}$	24.96	12.90	**12.63**	13.58
	VoiceCraft$_{\text{PAB}}$	7.53	15.14	13.53	21.70
	XTTS-v2$_{\text{PAB}}$	**4.16**	**4.31**	37.81	**8.32**
	viXTTS	4.23	5.17	37.81	12.54
MCD ↓	VALL-E$_{\text{PAB}}$	7.50	8.28	10.13	8.70
	VoiceCraft$_{\text{PAB}}$	6.69	7.98	10.27	9.15
	XTTS-v2$_{\text{PAB}}$	**6.30**	**7.81**	**9.85**	**8.34**
	viXTTS	7.47	8.48	10.54	8.71
RMSE$_{F0}$ ↓	VALL-E$_{\text{PAB}}$	226.55	246.88	267.80	**223.56**
	VoiceCraft$_{\text{PAB}}$	**214.66**	247.54	**259.46**	233.68
	XTTS-v2$_{\text{PAB}}$	216.44	**242.51**	290.77	228.81
	viXTTS	249.54	271.70	338.59	238.05
MOS ↑	Original	4.61 ± 0.17	4.63 ± 0.16	4.41 ± 0.14	4.66 ± 0.20
	VALL-E$_{\text{PAB}}$	3.96 ± 0.29	$\mathbf{4.04} \pm 0.28$	3.44 ± 0.21	3.75 ± 0.38
	VoiceCraft$_{\text{PAB}}$	4.16 ± 0.21	3.75 ± 0.29	$\mathbf{3.85} \pm 0.22$	$\mathbf{3.98} \pm 0.22$
	XTTS-v2$_{\text{PAB}}$	$\mathbf{4.20} \pm 0.20$	3.89 ± 0.21	2.79 ± 0.21	3.98 ± 0.29
	viXTTS	4.05 ± 0.23	3.85 ± 0.25	2.37 ± 0.24	3.48 ± 0.44
SMOS ↑	Original	4.23 ± 0.23	3.90 ± 0.32	3.87 ± 0.24	3.34 ± 0.47
	VALL-E$_{\text{PAB}}$	$\mathbf{3.77} \pm 0.24$	3.46 ± 0.29	$\mathbf{3.35} \pm 0.25$	3.20 ± 0.38
	VoiceCraft$_{\text{PAB}}$	3.64 ± 0.30	3.32 ± 0.35	3.25 ± 0.25	$\mathbf{3.41} \pm 0.36$
	XTTS-v2$_{\text{PAB}}$	3.55 ± 0.27	$\mathbf{3.56} \pm 0.29$	3.03 ± 0.23	3.39 ± 0.41
	viXTTS	2.88 ± 0.28	2.63 ± 0.32	2.48 ± 0.23	3.11 ± 0.43

Table 2: Test results of different TTS models. Our models, "VALL-E$_{\text{PAB}}$", "VoiceCraft$_{\text{PAB}}$" and "XTTS-v2$_{\text{PAB}}$" are obtained by training VALL-E, VoiceCraft, and XTTS-v2 on our PhoAudiobook training data, respectively. "PAB-S" and "PAB-U" refer to the PhoAudiobook "seen" and "unseen" speaker test sets, respectively. The viXTTS model is fine-tuned from the pre-trained XTTS-v2 using the entire viVoice dataset.

to rate the outputs for the in-distribution test sets (PAB-S, PAB-U) and 20 native speakers for the out-of-distribution test sets (VIVOS, viVoice), with all ratings on a scale from 1 to 5, using 0.5-point increments. To ensure fairness, we shuffle and anonymize the model names so that each listener is unaware of which model produces each sample.

4 Results

Table 2 presents the results obtained for our trained models and the baseline. It is clear that our XTTS-v2$_{\text{PAB}}$ consistently outperforms viXTTS across all metrics and test sets. For instance, on the viVoice set, XTTS-v2$_{\text{PAB}}$ achieves the best WER of 8.32, which is substantially lower than the 12.54 WER of viXTTS, even though viXTTS is tested on its own training data. Additionally, XTTS-v2$_{\text{PAB}}$ also produces substantially higher SMOS and RMSE$_{F0}$ scores compared to viXTTS in all test sets, indicating that the speech it generates more closely resembles the reference speaker. These results suggest that XTTS-v2$_{\text{PAB}}$ outputs more intelligible and natural-sounding speech that better captures the nuances of the target speaker's voice, for both long (PhoAudiobook and viVoice) and shorter (VIVOS) text inputs.

We observe a variation in the performance of different models across test sets. While VoiceCraft$_{\text{PAB}}$ and VALL-E$_{\text{PAB}}$ are less competent than XTTS-v2$_{\text{PAB}}$ on the test sets PAB-S, PAB-U and viVoice, they outperform XTTS-v2$_{\text{PAB}}$ on the VIVOS test set. Specifically, for PAB-S, PAB-U, and viVoice test sets, VALL-E$_{\text{PAB}}$ and VoiceCraft$_{\text{PAB}}$ underperform compared to XTTS-v2$_{\text{PAB}}$ in terms of WER, while achieving comparable results on other metrics such as MCD, RMSE$_{F0}$, MOS, and SMOS. However, on the VIVOS test set, XTTS-v2$_{\text{PAB}}$ and viXTTS perform significantly worse than VALL-E$_{\text{PAB}}$ and VoiceCraft$_{\text{PAB}}$ across all evaluation metrics. This indicates that VALL-E$_{\text{PAB}}$ and VoiceCraft$_{\text{PAB}}$ are more adept at handling short sentences, which are characteristic of the VIVOS test set. Upon manual inspection, we found that for short text inputs, XTTS-v2-based models – XTTS-v2$_{\text{PAB}}$ and viXTTS – often generate redundant or rambling speech at the end of the output. This suggests a potential architectural issue within the XTTS-v2 model itself, rather than a data-related problem, as both the viVoice dataset and the "augmented" PhoAudiobook training set contain short audio samples.

5 Conclusion

We have introduced PhoAudiobook, a comprehensive 941-hour high-quality dataset designed for Vietnamese text-to-speech (TTS) synthesis. Using this dataset, we conducted experiments with three leading zero-shot TTS models: VALL-E, VoiceCraft, and XTTS-v2. Our findings show that XTTS-v2 consistently outperforms its counterpart trained on the viVoice dataset across all metrics, highlighting the superiority of PhoAudiobook in enhancing model performance. Additionally, VALL-E and VoiceCraft demonstrate exceptional capability in handling short sentences.

Limitations

While models trained on PhoAudiobook show high performance on purely Vietnamese datasets, we have not evaluated their performance in code-switching scenarios where the input text includes both Vietnamese and English. Future research should investigate the models' ability to handle multilingual inputs to enhance their applicability in more diverse linguistic contexts.

Acknowledgments

This work was completed while all authors were at Movian AI, Vietnam. All datasets and models were downloaded, trained, and evaluated using Movian AI's resources.

We would like to express our sincere gratitude to Mr. Nguyen Nguyen for providing a toolkit that facilitated the process of downloading raw data from https://sachnoiviet.net.

References

Sercan Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. 2018. Neural Voice Cloning with a Few Samples. In *Proceedings of NeurIPS*.

James Betker. 2023. Better speech synthesis through scaling.

Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Göknar, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, and Julian Weber. 2024. Xtts: a massively multilingual zero-shot text-to-speech model. In *Proceedings of INTERSPEECH*.

Edresson Casanova, Julian Weber, Christopher D Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir A Ponti. 2022. YourTTS: Towards Zero-Shot Multi-Speaker TTS and Zero-Shot Voice Conversion for Everyone. In *Proceedings of ICML*.

Yuan-Jui Chen, Tao Tu, Cheng chieh Yeh, and Hung-Yi Lee. 2019. End-to-End Text-to-Speech for Low-Resource Languages by Cross-Lingual Transfer Learning. In *Proceedings of INTERSPEECH*.

Erica Cooper, Cheng-I Lai, Yusuke Yasuda, Fuming Fang, Xin Wang, Nanxin Chen, and Junichi Yamagishi. 2020. Zero-Shot Multi-Speaker Text-To-Speech with State-Of-The-Art Neural Speaker Embeddings. In *Proceedings of ICASSP*.

Alexandre Défossez. 2021. Hybrid Spectrogram and Waveform Source Separation. In *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*.

Thinh Le Phuoc Gia, Tuan Pham Minh, Hung Nguyen Quoc, Trung Nguyen Quoc, and Vinh Truong Hoang. 2024. viVoice: Enabling Vietnamese Multi-Speaker Speech Synthesis.

Alexander Gutkin, Linne Ha, Martin Jansche, Knot Pipatsrisawat, and Richard Sproat. 2016. TTS for Low Resource Languages: A Bangla Synthesizer. In *Proceedings of LREC*.

Rongjie Huang, Chunlei Zhang, Yongqi Wang, Dongchao Yang, Jinchuan Tian, Zhenhui Ye, Luping Liu, Zehan Wang, Ziyue Jiang, Xuankai Chang, Jiatong Shi, Chao Weng, Zhou Zhao, and Dong Yu. 2024. Make-A-Voice: Revisiting Voice Large Language Models as Scalable Multilingual and Multitask Learners. In *Proceedings of ACL*.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech. In *Proceedings of ICML*.

Thanh-Thien Le, Linh The Nguyen, and Dat Quoc Nguyen. 2024. PhoWhisper: Automatic Speech Recognition for Vietnamese. In *Proceedings of the ICLR 2024 Tiny Papers track*.

Jiaqi Li, Xueyao Zhang, Yuancheng Wang, Haorui He, Chaoren Wang, Li Wang, Huan Liao, Junyi Ao, Zeyu Xie, Yiqiao Huang, Junan Zhang, and Zhizheng Wu. 2025. Overview of the Amphion Toolkit (v0.2). *arXiv preprint*, arXiv:2501.15442.

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. 2019. Neural speech synthesis with transformer network. In *Proceedings of AAAI*.

Yanqing Liu, Ruiqing Xue, Lei He, Xu Tan, and Sheng Zhao. 2022. DelightfulTTS 2: End-to-End Speech Synthesis with Adversarial Vector-Quantized Auto-Encoders. In *Proceedings of INTERSPEECH*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Hieu-Thi Luong and Hai-Quan Vu. 2016. A non-expert Kaldi recipe for Vietnamese Speech Recognition System. In *Proceedings of WLSI/OIAF4HLT*, pages 51–55.

Florian Lux, Julia Koch, and Ngoc Thang Vu. 2022. Low-Resource Multilingual and Zero-Shot Multi-speaker TTS. In *Proceedings of AACL-IJCNLP*.

Phuong Pham Ngoc, Chung Tran Quang, and Mai Luong Chi. 2023. ADAPT-TTS: HIGH-QUALITY ZERO-SHOT MULTI-SPEAKER TEXT-TO-SPEECH ADAPTIVE-BASED FOR VIETNAMESE. *Journal of Computer Science and Cybernetics*, 39(2):159–173.

Puyuan Peng, Po-Yao Huang, Shang-Wen Li, Abdelrahman Mohamed, and David Harwath. 2024. VoiceCraft: Zero-Shot Speech Editing and Text-to-Speech in the Wild. In *Proceedings of ACL*.

Anh Pham, Khanh Linh Tran, Linh Nguyen, Thanh Duy Cao, Phuc Phan, and Duong A. Nguyen. 2024. Bud500: A Comprehensive Vietnamese ASR Dataset.

Viet Thanh Pham, Xuan Thai Hoa Nguyen, Vu Hoang, and Thi Thu Trang Nguyen. 2023. Vietnam-Celeb: a large-scale dataset for Vietnamese speaker recognition. In *Proceedings of INTERSPEECH*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of ICML*.

Mirco Ravanelli, Titouan Parcollet, Adel Moumen, Sylvain de Langen, Cem Subakan, Peter Plantinga, Yingzhi Wang, Pooneh Mousavi, Luca Della Libera, Artem Ploujnikov, Francesco Paissan, Davide Borra, Salah Zaiem, Zeyu Zhao, Shucong Zhang, Georgios Karakasidis, Sung-Lin Yeh, Pierre Champion, Aku Rouhe, Rudolf Braun, Florian Mai, Juan Zuluaga-Gomez, Seyed Mahed Mousavi, Andreas Nautsch, Ha Nguyen, Xuechen Liu, Sangeet Sagar, Jarod Duret, Salima Mdhaffar, Gaëlle Laperrière, Mickael Rouvier, Renato De Mori, and Yannick Estève. 2024. Open-Source Conversational AI with SpeechBrain 1.0. *Journal of Machine Learning Research*, 25(333).

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. FastSpeech: Fast, Robust and Controllable Text to Speech. In *Proceedings of NeurIPS*.

Simon Rouard, Francisco Massa, and Alexandre Défossez. 2023. Hybrid Transformers for Music Source Separation. In *Proceedings of ICASSP*.

Xu Tan, Jiawei Chen, Haohe Liu, Jian Cong, Chen Zhang, Yanqing Liu, Xi Wang, Yichong Leng, Yuanhao Yi, Lei He, Sheng Zhao, Tao Qin, Frank Soong, and Tie-Yan Liu. 2024. NaturalSpeech: End-to-End Text-to-Speech Synthesis With Human-Level Quality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(6):4234–4245.

VinBigData. 2023. VinBigData Shares 100-Hour Data for the Community.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2023. Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers. *arXiv preprint*, arXiv:2301.02111.

Tao Wang, Jianhua Tao, Ruibo Fu, Jiangyan Yi, Zhengqi Wen, and Rongxiu Zhong. 2020. Spoken Content and Voice Factorization for Few-Shot Speaker Adaptation. In *Proceedings of INTERSPEECH*.

Yihan Wu, Xu Tan, Bohan Li, Lei He, Sheng Zhao, Ruihua Song, Tao Qin, and Tie-Yan Liu. 2022. AdaSpeech 4: Adaptive Text to Speech in Zero-Shot Scenarios. In *Proceedings of INTERSPEECH*.

A Implementation details

For the VALL-E implementation, the first phase is the data processing stage. We fine-tune a model based on the Vietnamese Wav2Vec 2.0 large model[7] for the Vietnamese dialect recognition task using our in-house data. The model achieves an accuracy of approximately 95% for the dialect recognition task. We then apply this model to the PhoAudiobook. To accurately determine the regional dialect for each speaker and reduce inference time, we sample about 20 audios for each speaker and feed them into the model. We assign the dialect of each speaker based on the region with the highest number of predicted audios. Because the VALL-E model is trained based on the phoneme level, we use the "phonemizer" package to convert the text into phonemes based on the dialect of each speaker.[8] For the audio, we also use the Audio CodeC encoder to compress raw audio into discrete tokens. The second phase is model training. We employ 12 Transformer-decoder layers, each with 1024 hidden units and 16 attention heads. We use a batch size corresponding to a maximum duration of 40 audio seconds, a base learning rate of 0.05, and 4 gradient accumulation steps. Our model is trained on 8 A100-40GB GPUs. Our implementation is based on the customized GitHub repository that reproduces the idea from the VALL-E paper.[9] We make modifications to this repository for our specific language and data.

Since VoiceCraft takes phoneme representations as input, we first convert our data to phonemes using the "phonemizer" package. We then append the derived Vietnamese phonemes to the existing English vocabulary and expand the text embedding layer to accommodate Vietnamese phonemes. The 830M_TTSEnhanced checkpoint is a public VoiceCraft model fine-tuned with a text-to-speech objective and serves as our starting point. Following the author's implementation,[10] we fine-tune this model using the AdamW optimizer with a learning rate of $1e^{-5}$ and a batch size of $25,000$ tokens, which corresponds to approximately 8.3 minutes of audio. We train the model on 4 A100-40GB GPUs for 16 epochs.

XTTS-v2 employs BPE for text encoding. To

[7] https://huggingface.co/nguyenvulebinh/wav2vec2-large-vi

[8] https://github.com/bootphon/phonemizer

[9] https://github.com/lifeiteng/vall-e/tree/main

[10] https://github.com/jasonppy/VoiceCraft

adapt the training to Vietnamese data, we use the same Vietnamese token list employed by Gia et al. (2024). We follow the training recipes provided in the coqui's TTS repository and fine-tune the public XTTS-v2 checkpoint trained for 16 languages.[11] We extend the character and audio length limits to accommodate audio segments up to 20 seconds in duration for training data. We use the AdamW optimizer with a learning rate of $5e^{-6}$, a batch size of 4, and fine-tune the model on a single A100-40GB GPU for 18 epochs.

For all these 3 models, we select the model checkpoint that obtains the best loss on the PhoAudiobook validation set.

[11] https://github.com/coqui-ai/TTS

Can LLMs Generate High-Quality Test Cases for Algorithm Problems? TestCase-Eval: A Systematic Evaluation of Fault Coverage and Exposure

Zheyuan Yang[♣] **Zexi Kuang**[♢] **Xue Xia**[♡] **Yilun Zhao**[♠]

[♣]Tongji University [♢]Northeastern University [♡]HKUST [♠]Yale University

Abstract

We introduce TestCase-Eval, a new benchmark for systematic evaluation of LLMs in test-case generation. TestCase-Eval includes 500 algorithm problems and 100,000 human-crafted solutions from the Codeforces platform. It focuses on two pivotal tasks: (1) *Fault Coverage*, which measures how well LLM-generated test sets probe diverse input scenarios and cover a wide range of potential failure modes. (2) *Fault Exposure*, which evaluates whether LLMs can craft a tailored test input that reveals a specific incorrect code implementation. We provide a comprehensive assessment of 19 state-of-the-art open-source and proprietary LLMs on TestCase-Eval, offering insights into their strengths and limitations in generating effective test cases for algorithm problems.

🤗 **Data** `TestCase-Eval`
Code `FlowRays/TestCase-Eval`

1 Introduction

Algorithmic problem-solving is fundamental to computational fields such as software engineering, data science, and competitive programming (Jimenez et al., 2023; Huang et al., 2023; Jain et al., 2024; Yu et al., 2024a; El-Kishky et al., 2025). The correctness and robustness of algorithmic solutions hinge on the quality of test suites—carefully designed inputs that uncover edge cases, corner conditions, performance limitations, and common failure scenarios (Austin et al., 2021; Hendrycks et al., 2021; Li et al., 2022). Traditionally, crafting such test cases requires significant domain expertise and manual effort. With the rapid advancement of LLMs capable of sophisticated code generation, a crucial question arises: **Can LLMs generate high-quality test cases that match or surpass those designed by human experts?**

We introduce TestCase-Eval, a comprehensive benchmark for systematically evaluating LLMs in test-case generation for algorithmic problems. It comprises 500 up-to-date algorithm problems and 100,000 corresponding real-human crafted solutions, both sourced from the Codeforces platform. As illustrated in Figure 1, TestCase-Eval features two core tasks, each targeting a crucial aspect of test-case quality: (1) *Fault Coverage*, which evaluates whether LLM-generated test cases effectively explore diverse input scenarios, including edge cases and boundary conditions, to expose various types of incorrect solutions. (2) *Fault Exposure*, which evaluates whether an LLM can generate a targeted test input that successfully exposes the flaws in a specific given incorrect solution.

We conduct an extensive evaluation on TestCase-Eval, covering 19 frontier open-source and proprietary LLMs. Our experimental results demonstrate that TestCase-Eval presents a significant challenge, with even top-performing models like Qwen3-32B scoring only 43.8% on the Fault Exposure task—far below human expert performance (93.3%). These findings underscore the inherent difficulty of TestCase-Eval. Furthermore, our in-depth analysis of reasoning LLMs, CoT reasoning, and model performance across different programming languages and error types offers valuable insights for future advancements in the field.

2 Related Work

Prior works on LLM-based test-case generation follows two main directions: (1) Enhancing code generation via self-debugging, where models iteratively refine solutions by generating and analyzing test cases (Chen et al., 2022; Zhang et al., 2023; Shinn et al., 2023; Jiao et al., 2024; Zeng et al., 2025). (2) Improving code evaluation by leveraging LLMs to generate diverse test cases, as used in recent code generation evaluation benchmarks (Liu et al., 2023; Du et al., 2024; Yu et al., 2024b; Jain et al., 2024). Despite these advancements, a sys-

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1050–1063
July 27 - August 1, 2025

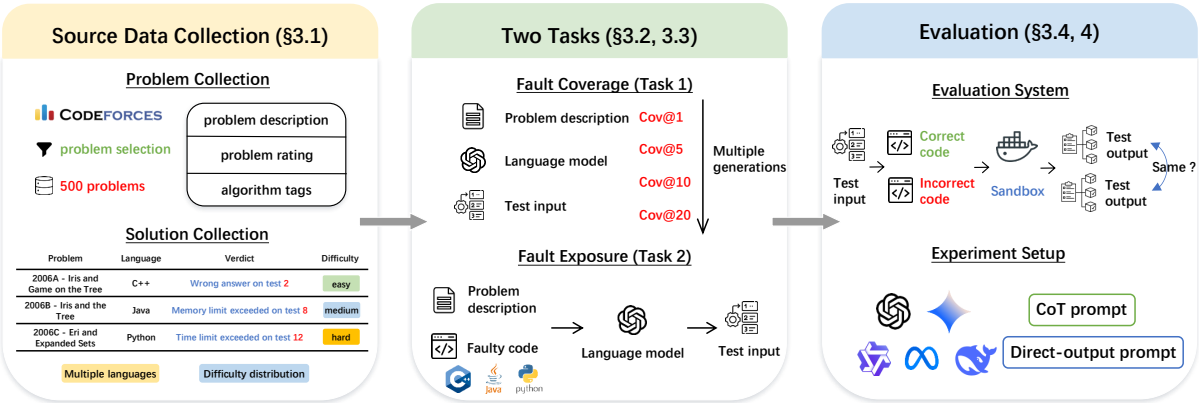


Figure 1: An overview of TestCase-Eval and the research pipeline in this study.

tematic study on the standalone capability of LLMs in test-case generation remains an open challenge. A closely related benchmark for algorithmic problem test-case generation is TestEval (Wang et al., 2024), which evaluates test generation for LeetCode problems but relies on traditional Line and Branch Coverage assessment (illustrated in Appendix A.1), which may be inadequate for algorithmic problem settings. For instance, the 6.7B DeepSeek-Coder achieves over 90% in TestEval, with top models nearing 100%. Our work shifts the focus to more challenging CodeForces competition problems and introduces two novel and challenging tasks. Even state-of-the-art models achieve only around 40% on the task of Fault Exposure.

3 TestCase-Eval Benchmark

This section discusses the TestCase-Eval benchmark construction and task settings, as illustrated in Figure 1.

3.1 Source Data Collection

We first outline the data collection process, which involves collecting *problems* with corresponding correct and incorrect *human-written solutions*.

Problem Collection. We collect algorithmic problems from Codeforces contests held between January 1, 2024, and December 30, 2024. This time frame falls outside the pretraining period of most existing foundation models, reducing potential data memorization concerns. Our goal is to curate a dataset that (1) includes a substantial number of incorrect submissions and (2) ensures accurate offline evaluation. To achieve this, we apply a series of filtering steps. First, we exclude problems requiring special judge functionalities (detailed in Appendix B.3), as these allow multiple valid outputs for a single input, often leading to unreliable evaluations. Next, we verify each problem by running ten correct human-written solutions sourced from Codeforces, ensuring they consistently produce identical outputs for the same test inputs. Additionally, we remove problems with fewer than 1,000 online incorrect submissions to ensure a diverse range of mistakes. After this selection process, our final dataset includes a total of 500 problems. Figure 3 in Appendix presents the distribution of problem difficulty ratings.

Human-written Solution Collection. For each problem in TestCase-Eval, we collect 200 incorrect submissions (*i.e.,* human-written solutions along with their evaluation outcomes) from the Codeforces platform. The platform provides detailed error types for each incorrect submission, such as "Memory Limit Exceeded", "Time Limit Exceeded", "Runtime Error", and "Wrong Answer". Additionally, it specifies the test case index where the error occurred (*e.g.,* "Wrong answer on test 5"), with higher indices generally indicating more complex or inherent errors. Leveraging this information, we categorize incorrect submissions into three difficulty levels: *Easy*, *Medium*, and *Hard*. (Detailed difficulty definition and distribution are illustrated in Appendix B.2) To ensure diversity, we select code written in three widely used programming languages: C++, Python, and Java. We begin by crawling the submission logs for each problem, which contain the complete history of contestant submissions. These logs are carefully filtered based on test case failures and programming language criteria, resulting in a comprehensive dataset of 100,000 submissions. (Detailed dataset collection and sampling pipeline are illustrated in Ap-

pendix B.2)

We next outline the construction process for the two evaluation tasks in TestCase-Eval.

3.2 Fault Coverage Evaluation (Task 1)

This task assesses the LLM's ability to generate comprehensive test inputs that effectively detect faulty code implementations. Specifically, given the description of an algorithmic problem, the LLM must thoroughly understand the problem and generate a specified number of test cases that maximize coverage of incorrect solution scenarios. Let $\mathcal{T}_N = \{t_1, t_2, \ldots, t_N\}$ represent the set of N test inputs generated by the LLM. For each test input t_i, let $\mathcal{F}(t_i)$ denote the subset of incorrect submissions it detects, drawn from the complete set of incorrect solutions $\mathcal{F}_{\text{total}}$, the final score for this task is defined as the coverage rate of incorrect solutions when generating N test inputs:

$$Cov@N = \frac{\left|\bigcup_{i=1}^{N} \mathcal{F}(t_i)\right|}{|\mathcal{F}_{\text{total}}|}$$

It quantifies the LLM's effectiveness in generating diverse and impactful test cases that expose incorrect implementations.

3.3 Fault Exposure Evaluation (Task 2)

This task is inspired by the hacking phase in CodeForces competitions, where participants analyze others' solutions and attempt to "hack" them by providing inputs that reveal flaws in the code. The goal is to assess the LLM's ability to understand both the problem and the specific errors present in the faulty implementation. Given the description of an algorithmic problem and a single faulty code implementation f_i within the sampled set $\mathcal{F}$ (a strategically sampled subset of $\mathcal{F}_{\text{total}}$), Task 2 requires the LLM to generate a single test input t_i to exploit the fault. The *Fault Exposure Rate* is computed as:

$$\text{Fault Exposure Rate} = \frac{1}{|\mathcal{F}|} \sum_{f_i \in \mathcal{F}} e(f_i, t_i), \text{ where}$$

$$e(f_i, t_i) = \begin{cases} 1, & \text{if } t_i \text{ successfully exposes fault in } f_i, \\ 0, & \text{otherwise.} \end{cases}$$

It measures both general test case generation capabilities and targeted fault detection performance.

4 Experiment

4.1 Experiment Setup

We evaluate 11 series of open-source models, including Qwen2.5 (Yang et al., 2024b) and Qwen2.5-Coder (Hui et al., 2024), Qwen3 (Yang et al., 2025), QwQ (Team, 2025), Llama-3.1&3.3 (Meta, 2024), Gemma-3 (Team et al., 2025), DeepSeek-R1 (DeepSeek-AI et al., 2025), Mistral-Small (Jiang et al., 2023), Codestral (Team, 2024), and SeedCoder (Seed, 2025). We also evaluate two series of proprietary models, including GPT-4o (OpenAI, 2024) and GPT-4.1 (OpenAI, 2025). Appendix C.1 details the parameter settings and model configurations. We evaluate the models with both **Direct Output** and **Chain-of-Thought** prompts, with prompting examples presented in Appendix C.2. We utilize the **sandbox environment** from ExecEval (Khan et al., 2023) for code execution and test input evaluation, ensuring secure execution and reliable assessment of results. To approximate **human-expert-level performance** on TestCase-Eval, we randomly sampled 20 problems from the dataset. Two human experts, with Codeforces ratings of 2080 and 2237, independently completed both Task 1 and Task 2 for each problem. Their performance was then averaged to obtain the final assessment.

4.2 Experimental Results and Analysis

Table 1 illustrates the model performance on TestCase-Eval. Our key findings are as follows:

TestCase-Eval presents substantial challenges for current models. The TestCase-Eval benchmark is highly challenging, as evidenced by the significant performance gap between models and human experts on both tasks. This gap is particularly pronounced in Task 2 (Fault Exposure), where human experts achieve a 93.3% fault exposure rate, more than double the best-performing model, Qwen3-32B (43.8%). While Task 1 (Fault Coverage) also shows a considerable gap, models achieve relatively higher scores, suggesting that generating a broad set of test cases is more tractable than triggering specific code flaws. Furthermore, we observe that Task 2 yields more stable and reproducible results across multiple evaluation runs, whereas Task 1 scores exhibit higher variance, likely due to the stochastic nature of generating a diverse set of test inputs.

Models	T1: Fault Coverage				T2: Fault Exposure			
	c@1	c@5	c@10	c@20	Easy	Med.	Hard	**Ovr.**
Human Expert	56.2	85.7	93.5	97.2	95.0	92.5	91.8	93.3
GPT-4.1	45.3	67.5	74.1	80.0	42.9	34.3	30.3	36.5
GPT-4.1-mini	38.8	63.2	68.5	72.6	39.2	32.4	27.4	33.6
GPT-4o	36.4	60.3	69.7	73.5	37.5	30.5	25.2	31.7
Qwen3-8B	46.2	78.5	87.9	92.1	48.6	39.8	33.1	41.3
Qwen3-32B	50.8	82.3	92.6	95.7	52.7	42.5	33.2	43.8
R1-Distill-Qwen-32B	31.9	65.3	75.6	82.6	48.7	39.7	33.9	41.6
QwQ-32B	37.3	58.9	67.6	78.3	49.4	38.0	30.2	40.2
Qwen2.5-7B	38.6	65.4	73.0	79.1	39.8	34.2	29.5	35.0
Qwen2.5-Coder-7B	36.7	63.2	70.5	76.5	37.2	33.1	29.9	33.7
Qwen2.5-32B	44.4	70.9	79.6	88.4	38.8	30.4	25.5	32.3
Qwen2.5-Coder-32B	35.8	66.7	81.8	89.7	40.5	34.0	27.3	34.6
Qwen2.5-72B	38.2	57.8	65.2	73.1	33.6	27.5	24.2	29.0
Llama-3.1-70B	47.8	75.4	84.8	90.9	37.9	33.5	30.5	34.3
Llama-3.3-70B	43.2	72.5	81.2	88.6	33.8	27.9	25.2	29.5
Mistral-Small-24B	35.5	71.9	80.4	88.3	37.4	31.9	28.4	33.1
Codestral-22B	34.8	68.8	87.4	90.8	34.9	28.2	26.4	30.3
Gemma-3-12B	30.4	54.6	61.0	65.3	35.7	31.9	33.3	33.8
Gemma-3-27B	32.4	55.6	64.1	70.7	34.3	28.3	28.3	30.7
Seed-Coder-8B	30.7	63.2	75.6	87.4	34.5	28.1	25.5	29.9

Table 1: Performance of the evaluated LLMs with CoT reasoning on TestCase-Eval. For Task 1, we report Coverage@N; for Task 2, we report fault exposure rate.

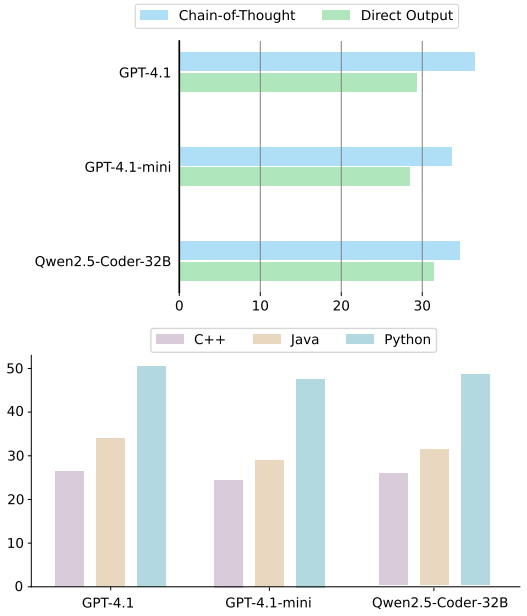


Figure 2: (Top) Performance comparison between CoT prompting and direct-output prompting for Task 2. (Bottom) Overall model performance using CoT prompting across C++, Java, and Python in Task 2.

Open-source models compete with or surpass proprietary counterparts. Our results indicate that leading open-source models are highly competitive. In Task 1 (Fault Coverage), several open-source models, including Qwen3-32B (50.8 cov@1) and Llama-3.1-70B (47.8 cov@1), outperform the best proprietary model, GPT-4.1 (45.3 cov@1). The trend continues at higher N values, where Qwen3-32B's cov@20 score of 95.7 significantly surpasses GPT-4.1's 80.0. In the more reasoning-intensive Task 2, while the Qwen3 series leads, GPT-4.1 shows strong performance with a 36.5% fault exposure rate, outperforming all other general-purpose open-source models like Llama-3.1-70B (34.3%). This highlights a competitive landscape where proprietary models do not hold a universal advantage in our benchmark.

Reasoning LLMs outperform general-purpose LLMs on both tasks. Reasoning-oriented models, such as the Qwen3 series, demonstrate superior performance on both tasks. Notably, Qwen3-32B achieves the highest scores in Task 1 across all metrics (e.g., 50.8 cov@1 and 95.7 cov@20), clearly surpassing strong general-purpose models such as Llama-3.1-70B, as well as proprietary models like GPT-4.1. This performance gap becomes even more pronounced in Task 2, which demands deeper analytical capabilities. Qwen3-32B and R1-Distill-Qwen-32B attain the top two fault exposure rates, at 43.8% and 41.6% respectively, with a substantial margin over all other evaluated models. These results suggest that reasoning models excel because they are better equipped to analyze algorithmic problem descriptions, systematically identify possible fault patterns, and generate high-quality test inputs.

CoT prompts vs direct-output prompts. Our experiments reveal that CoT prompting significantly outperforms direct-output prompting in generating test cases (Figure 2). This advantage stems from CoT's structured reasoning process, which guides the model through intermediate steps before arriving at the final output. Such an approach is particularly beneficial for the complex tasks in TestCase-Eval, where systematic thinking is crucial. When generating test cases for fault exposure, CoT prompting led to more effective fault detection, especially in challenging edge cases. This suggests that models benefit from explicit reasoning steps, as they help decompose intricate problems and improve fault exposure rate in nuanced scenarios.

Comparison of fault exposure results across different programming languages. In Fault Exposure task, model performance varies across pro-

gramming languages. As shown in Figure 2, models generally achieve higher fault exposure rates on Python solutions, likely due to Python's dynamic typing, flexible syntax, and interpreted execution, which facilitate the generation of diverse test cases that reveal faults. In contrast, C++ and Java exhibit lower fault exposure rates, possibly due to their strict type enforcement, manual memory management, and compiled execution, which can limit the likelihood of generating inputs that reveal subtle faults. Addressing these differences through targeted adaptation could help improve fault detection across a wider range of programming languages.

Performance breakdown to four major error types. Table 2 provides a detailed breakdown of model performance on task 2 across four major error types: Wrong Answer (WA), Runtime Error (RE), Time Limit Exceeded (TLE), and Memory Limit Exceeded (MLE). A clear trend emerges from the data: models generally demonstrate stronger capabilities in detecting logical and execution-related faults (WA and RE) compared to resource-based faults (TLE and MLE). This suggests that current LLMs are generally better at detecting logical or edge-case errors than time or memory inefficiencies.

Notably, the best-performing models, Qwen3-32B and R1-Distill-Qwen-32B, exhibit significantly higher accuracy on Wrong Answer type. Given that WA constitutes the largest proportion of error types, their superior performance in this category is the primary driver of their high overall accuracy (43.8% and 41.6%, respectively). This finding underscores that the strength of these advanced reasoning models lies in their enhanced ability to construct precise test cases that target and expose logical inconsistencies within code.

5 Conclusion

We introduce TestCase-Eval, a new benchmark for evaluating LLMs in test-case generation for algorithmic problems, with a focus on Fault Coverage and Fault Exposure. Our results show that all the evaluated LLMs struggle with harder faults, highlighting the challenge of automated test-case generation. Although CoT prompting enhances performance, a substantial gap remains between frontier LLMs and human experts. These findings emphasize the need for further research to enhance LLMs' capabilities in generating high-quality test cases and their practical application in software testing.

Models	**WA**	**RE**	**TLE**	**MLE**	**Ovr.**
GPT-4.1	42.0	35.4	20.9	25.1	36.5
GPT-4.1-mini	39.3	32.0	17.4	24.6	33.6
GPT-4o	37.1	29.0	16.4	25.1	31.7
Qwen3-8B	48.0	39.0	22.8	26.9	41.3
Qwen3-32B	52.2	38.7	21.2	22.3	43.8
R1-Distill-Qwen-32B	48.0	37.8	23.9	30.3	41.6
QwQ-32B	49.1	35.1	16.4	14.9	40.2
Qwen2.5-7B	38.7	37.4	23.6	31.4	35.0
Qwen2.5-Coder-7B	35.5	39.8	26.5	36.0	33.7
Qwen2.5-32B	36.8	34.7	18.0	28.6	32.3
Qwen2.5-Coder-32B	38.9	38.3	21.0	26.9	34.6
Qwen2.5-72B	33.0	30.5	16.8	20.6	29.0
Llama-3.1-70B	36.5	33.9	27.7	36.6	34.3
Llama-3.3-70B	32.9	30.2	19.3	24.6	29.5
Mistral-Small-24B	35.8	37.4	23.1	40.0	33.1
Codestral-22B	33.1	35.0	20.3	31.4	30.3
Gemma-3-12B	35.8	35.1	27.7	30.9	33.8
Gemma-3-27B	33.5	33.3	22.3	21.7	30.7
Seed-Coder-8B	32.8	31.7	20.5	31.4	29.9

Table 2: Performance breakdown of evaluated LLMs on task 2 (fault exposure), reported by four error types.

Limitations

While TestCase-Eval provides a comprehensive benchmark for evaluating LLMs in test-case generation, several limitations warrant further investigation: (1) Our evaluation primarily emphasizes quantitative performance indicators, such as fault coverage and exposure rates, which might not capture the nuanced failure modes that may arise in LLM-generated test cases. Future work could include more detailed error analyses to uncover specific failure patterns and model weaknesses. (2) The difficulty levels (Easy, Medium, Hard) in TestCase-Eval are determined by the test case index where an incorrect solution first fails. While this provides a reasonable estimate of error complexity, it does not explicitly categorize the types of mistakes. (3) Our benchmark focuses on correctness-based faults and does not systematically test performance bottlenecks (*e.g.,* time limit exceeded, memory limit exceeded). Although some incorrect submissions fail due to resource constraints, we do not explicitly assess whether LLMs generate test cases that effectively expose computational complexity flaws. (4) While TestCase-Eval targets test-case generation, practical software testing also requires identifying the root cause and location of bugs. Future work could extend this to more holistic debugging and fault localization tasks.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. 2021. Program synthesis with large language models. *ArXiv*, abs/2108.07732.

Bei Chen, Fengji Zhang, A. Nguyen, Daoguang Zan, Zeqi Lin, Jian-Guang Lou, and Weizhu Chen. 2022. Codet: Code generation with generated tests. *ArXiv*, abs/2207.10397.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bing-Li Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dong-Li Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Jiong Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, M. Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shao-Kang Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wen-Xia Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyu Jin, Xi-Cheng Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yi Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yu-Jing Zou, Yujia He, Yunfan Xiong, Yu-Wei Luo, Yu mei You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yao Li, Yi Zheng, Yuchen Zhu, Yunxiang Ma, Ying Tang, Yukun Zha, Yuting Yan, Zehui Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhen guo Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zi-An Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Mingzhe Du, Anh Tuan Luu, Bin Ji, Qian Liu, and See-Kiong Ng. 2024. Mercury: A code efficiency benchmark for code large language models. In *Neural Information Processing Systems*.

OpenAI Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub W. Pachocki, Jerry Tworek, Lorenz Kuhn, Lukasz Kaiser, Mark Chen, Max Schwarzer, Mostafa Rohaninejad, Nat McAleese, o3 contributors, Oleg Murk, Rhythm Garg, Rui Shu, Szymon Sidor, Vineet Kosaraju, and Wenda Zhou. 2025. Competitive programming with large reasoning models.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Xiaodong Song, and Jacob Steinhardt. 2021. Measuring coding challenge competence with apps. *ArXiv*, abs/2105.09938.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2023. Mlagentbench: Evaluating language agents on machine learning experimentation. In *International Conference on Machine Learning*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, An Yang, Rui Men, Fei Huang, Shanghaoran Quan, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. 2024. Qwen2.5-coder technical report. *ArXiv*, abs/2409.12186.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida I. Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *ArXiv*, abs/2403.07974.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Fangkai Jiao, Geyang Guo, Xingxing Zhang, Nancy F. Chen, Shafiq Joty, and Furu Wei. 2024. Preference optimization for reasoning with pseudo feedback. *ArXiv*, abs/2411.16345.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. Swe-bench: Can language models resolve real-world github issues? *ArXiv*, abs/2310.06770.

Mohammad Abdullah Matin Khan, M Saiful Bari, Do Xuan Long, Weishi Wang, Md. Rizwan Parvez, and Shafiq R. Joty. 2023. xcodeeval: A large scale multilingual multitask benchmark for code understanding, generation, translation and retrieval. *ArXiv*, abs/2303.03004.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom, Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de, Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey, Cherepanov, James Molloy, Daniel Jaymin Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de, Freitas, Koray Kavukcuoglu, and Oriol Vinyals. 2022. Competition-level code generation with alphacode. *Science*, 378:1092 – 1097.

Jiawei Liu, Chun Xia, Yuyao Wang, and Lingming Zhang. 2023. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *ArXiv*, abs/2305.01210.

Meta. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

OpenAI. 2024. Hello gpt-4o.

OpenAI. 2025. Introducing gpt-4.1 in the api.

ByteDance Seed. 2025. Seed-coder: Let the code model curate data for itself.

Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Neural Information Processing Systems*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. 2025. Gemma 3 technical report.

Mistral AI Team. 2024. Codestral: Hello, world!

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Wenhan Wang, Chenyuan Yang, Zhijie Wang, Yuheng Huang, Zhaoyang Chu, Da Song, Lingming Zhang, An Ran Chen, and Lei Ma. 2024. Testeval: Benchmarking large language models for test case generation. *ArXiv*, abs/2406.04531.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan

Qiu. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Ke-Yang Chen, Kexin Yang, Mei Li, Min Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yunyang Wan, Yunfei Chu, Zeyu Cui, Zhenru Zhang, and Zhi-Wei Fan. 2024a. Qwen2 technical report. *ArXiv*, abs/2407.10671.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. 2024b. Qwen2.5 technical report. *ArXiv*, abs/2412.15115.

Zhaojian Yu, Yilun Zhao, Arman Cohan, and Xiao-Ping Zhang. 2024a. Humaneval pro and mbpp pro: Evaluating large language models on self-invoking code generation.

Zhaojian Yu, Yilun Zhao, Arman Cohan, and Xiao-Ping Zhang. 2024b. Humaneval pro and mbpp pro: Evaluating large language models on self-invoking code generation. *ArXiv*, abs/2412.21199.

Huaye Zeng, Dongfu Jiang, Haozhe Wang, Ping Nie, Xiaotong Chen, and Wenhu Chen. 2025. Acecoder: Acing coder rl via automated test-case synthesis.

Kexun Zhang, Danqing Wang, Jingtao Xia, William Yang Wang, and Lei Li. 2023. Algo: Synthesizing algorithmic programs with generated oracle verifiers. *ArXiv*, abs/2305.14591.

A Related Work

A.1 Line and Branch Coverage

In the realm of software testing, two pervasive metrics are Line Coverage and Branch Coverage. These are often used to evaluate the adequacy of test cases in executing program code.

Specifically, **Line Coverage** measures the percentage of lines of code that have been executed by a set of test cases. It provides insight into which lines of the codebase are actually executed during testing, aiming to ensure that all parts of the code are tested at least once. **Branch Coverage** takes a more granular approach by focusing on the control structures within the code, such as if statements and switch cases. It evaluates whether each possible branch (i.e., each path through a control structure) has been executed. This metric ensures that all possible execution paths are tested.

While these metrics are invaluable in traditional software testing, they fall short in the context of competitive algorithm problems for several reasons: (1) Algorithm Complexity and Diversity: Competitive algorithm problems often involve complex data structures and intricate algorithmic logic that cannot be fully represented by simple line or branch execution. The focus is on the correctness and efficiency of the algorithm, rather than merely executing each line or branch of code. (2) Outcome-Oriented Nature: The primary goal in algorithm competitions is to solve problems correctly and efficiently, not just to achieve high code coverage. An algorithm may achieve high line and branch coverage but still fail to solve the problem correctly or efficiently. (3) Diversity of Test Cases: Algorithm competition problems require testing against a wide variety of edge cases and specific inputs. The generation and evaluation of these test cases extend beyond the scope of simple line and branch coverage metrics, which may not adequately reflect the comprehensiveness of the test cases in ensuring algorithmic correctness and robustness. Traditional Line and Branch Coverage metrics may be inadequate in the context of algorithmic problems. Therefore, we propose two new tasks, Fault Coverage and Fault Exposure, along with corresponding evaluation metrics.

B TestCase-Eval Benchmark

We provide a detailed explanation of the problems and human-written solutions within the dataset.

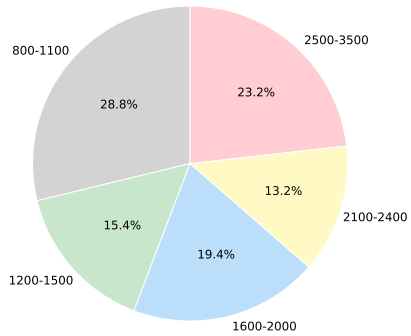


Figure 3: Distribution of Problem Difficulty Levels.

B.1 Problems

Each problem sourced from the Codeforces platform comprises several key elements: 1) title, 2) time limit, 3) memory limit, 4) problem description, 5) input format, 6) output format, 7) test case examples, and 8) optional note. We utilize all of this data to form the `problem_description` string, which acts as the input for the LLM.

Additionally, we analyze the distribution of problem difficulty ratings, which is illustrated in Figure Figure 3.

B.2 Human-written Solutions

To ensure both the representativeness and diversity of error patterns in our benchmark, we developed a comprehensive dataset collection and sampling pipeline.

For each selected Codeforces problem, we first crawled the complete submission logs to collect a representative set of user-submitted incorrect solutions. Specifically, for each problem, we initially sampled 100 incorrect solutions for each of the three major programming languages (C++, Python, and Java), resulting in a preliminary pool. We then applied multiple rounds of filtering and cleaning to ensure quality and diversity. This process yielded a final set of 118,611 human-written incorrect solutions across 500 algorithmic problems, amounting to an average of 237 solutions per problem. These submissions reflect genuine programming errors from a diverse pool of users, capturing a broad spectrum of error types and difficulty levels observed in real-world programming scenarios.

We imposed strict criteria on the sampled solutions: each must be semantically valid and executable, passing compilation and basic test cases without syntax or runtime errors, and failing only under specific, non-trivial input conditions. This design ensures our benchmark targets input-sensitive

faults—precisely those that require sophisticated and diverse test input generation to detect.

To construct a manageable yet representative subset for Task 2, we performed stratified sampling for each problem. We began by analyzing the distribution of incorrect solutions by error type and difficulty, which we defined based on the index of the first failed test case. Guided by this analysis, we sampled 20 incorrect solutions per problem, ensuring balanced representation across three major programming languages and maintaining proportional coverage of both error types and difficulty levels.

Codeforces problems typically include between a dozen and over two hundred test cases, each comprising a set of inputs and expected outputs. For a given submission, the verdict "Wrong answer on test 5" indicates that the solution passed the first four test cases but failed on the fifth. The index of the first failed test case thus serves as a crucial signal for assessing both the difficulty of a solution and the effectiveness of generated test cases.

Based on this index, we categorize solutions into three levels of difficulty: *Easy*, *Medium*, and *Hard*. Specifically, we sort all human-written solutions for each problem by the index of the first error, assigning the bottom 40% to *Easy*, the middle 30% to *Medium*, and the top 30% to *Hard*.

B.3 Special Judge

In competitive programming platforms like Codeforces, certain problems permit multiple valid test outputs for a single test input. To validate such outputs, a special judge is employed. This custom code evaluates the correctness of each output, as straightforward comparison to a reference output is inadequate due to the problem's complexity. Figure 4 illustrates a problem that necessitates a special judge.

For accurate offline evaluation, we excluded all problems requiring a special judge. Such problems can cause inconsistent assessments since they allow multiple correct outputs for the same input.

B.4 Multiple Test Cases

In competitive programming platforms (especially Codeforces), multiple test cases within a single test input are a standard feature. As illustrated in Figure 4, problem input specifications often begin with instructions such as "Each test contains multiple test cases" emphasizing the expectation that solutions correctly process a batch of cases in one execution. Through manual inspection of our dataset, we verified that 439 out of the 500 problems inherently require handling multiple test cases. To ensure comprehensive test coverage, we designed our prompt in Appendix C.2 to explicitly guide LLMs in generating diverse test cases in a single test input.

B. Kevin and Geometry

time limit per test: 1 second
memory limit per test: 256 megabytes

Kevin has n sticks with length $a_1, a_2, \ldots, a_n$.

Kevin wants to select 4 sticks from these to form an isosceles trapezoid* with a positive area. Note that rectangles and squares are also considered isosceles trapezoids. Help Kevin find a solution. If no solution exists, output -1.

* An isosceles trapezoid is a convex quadrilateral with a line of symmetry bisecting one pair of opposite sides. In any isosceles trapezoid, two opposite sides (the bases) are parallel, and the two other sides (the legs) are of equal length.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($4 \le n \le 2 \cdot 10^5$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^8$).

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output 4 integers — the lengths of sticks. If no solution exists, output -1.

If there are multiple solutions, print any of them. → This indicts the problem needs a special judge

Example

input (Copy)

```
7
4
5 5 5 10
4
10 5 10 5
4
1 2 3 4
4
1 1 1 3
6
4 2 1 5 7 1
6
10 200 30 300 30 100
4
100000000 100000000 1 2
```

output (Copy)

```
5 5 5 10
5 5 10 10
-1
-1
1 1 4 5
-1
100000000 100000000 1 2
```

Note

In the first test case, you can form an isosceles trapezoid with bases of length 5 and 10, and two legs of length 5.

In the second test case, you can form an isosceles trapezoid with two bases of length 5 and two legs of length 10. A rectangle is considered an isosceles trapezoid here.

In the third test case, there are no sticks with the same length. It's impossible to form an isosceles trapezoid.

In the fourth test case, it's impossible to form an isosceles trapezoid with a positive area.

Figure 4: An example of a problem that needs a special judge.

C Experiment Setup

C.1 Evaluated Model Configuration

Table 3 details the configuration of each evaluated model. Across all experiments, the temperature is set to 1.0 to ensure diversity in the LLM-generated test cases. The maximum output length is generally configured to 2048 tokens, which suffices for most standard models. However, for reasoning models like QwQ-32B and R1-Distill-Qwen-32B, this maximum output length is extended to 18,000 tokens to accommodate their long CoT reasoning mechanisms. All inference processes are conducted on two NVIDIA A100-80G GPUs.

C.2 CoT and Direct Output Prompts

Model	Citation	Version
GPT-4.1	OpenAI (2025)	gpt-4.1-2025-04-14
GPT-4.1-mini	OpenAI (2025)	gpt-4.1-mini-2025-04-14
GPT-4o	OpenAI (2024)	gpt-4o-2024-11-20
Qwen3-8B	Yang et al. (2025)	Qwen/Qwen3-8B
Qwen3-32B	Yang et al. (2025)	Qwen/Qwen3-32B
R1-Distill-Qwen-32B	DeepSeek-AI et al. (2025)	deepseek-ai/DeepSeek-R1-Distill-Qwen-32B
QwQ-32B	Team (2025)	Qwen/QwQ-32B
Qwen2.5-7B	Yang et al. (2024a)	Qwen/Qwen2.5-7B-Instruct
Qwen2.5-Coder-7B	Hui et al. (2024)	Qwen/Qwen2.5-Coder-7B-Instruct
Qwen2.5-32B	Yang et al. (2024b)	Qwen/Qwen2.5-32B-Instruct
Qwen2.5-Coder-32B	Hui et al. (2024)	Qwen/Qwen2.5-Coder-32B-Instruct
Qwen2.5-72B	Yang et al. (2024b)	Qwen/Qwen2.5-72B-Instruct
Llama-3.1-70B	Meta (2024)	meta-llama/Llama-3.1-70B-Instruct
Llama-3.3-70B	Meta (2024)	meta-llama/Llama-3.3-70B-Instruct
Mistral-Small-24B	Jiang et al. (2023)	mistralai/Mistral-Small-24B-Instruct-2501
Codestral-22B	Team (2024)	mistralai/Codestral-22B-v0.1
Gemma-3-12B	Team et al. (2025)	google/gemma-3-12b-it
Gemma-3-27B	Team et al. (2025)	google/gemma-3-27b-it
Seed-Coder-8B	Seed (2025)	ByteDance-Seed/Seed-Coder-8B-Instruct

Table 3: Model List.

The Chain-of-Thought Prompt in Task1

Task:
Generate a challenging test input for the algorithm problem:
{problem_description}

Instructions:
- Focus on edge cases or scenarios that maximize the failure probability in faulty solutions.
- Due to the output length limit, you should generate a small-scale test input that is complete and valid.
- Output the test input directly, not code to generate it.

Output format:

```plaintext
{test input}
```

Think step by step.

Figure 5: The Chain-of-Thought prompt used in Task1.

The Direct Output prompt in Task1

Task:
Generate a challenging test input for the algorithm problem:
{problem_description}

Instructions:
- Focus on edge cases or scenarios that maximize the failure probability in faulty solutions.
- Due to the output length limit, you should generate a small-scale test input that is complete and valid.
- Output the test input directly, not code to generate it.

Output format:

```plaintext
{test input}
```

Only output the test input, no explanations.

Figure 6: The Direct Output prompt used in Task1.

The Chain-of-Thought prompt in Task2

Task:
Generate a challenging test input that exposes the bug in the buggy code of the algorithm problem:
Algorithm Problem: {problem_description}
Buggy Code: {buggy_code}

Instructions:
- Focus on edge cases or scenarios that maximize the failure probability in faulty solutions.
- Due to the output length limit, you should generate a small-scale test input that is complete and valid.
- Output the test input directly, not code to generate it.

Output format:

```plaintext
{test input}
```

Think step by step.

Figure 7: The Chain-of-Thought prompt used in Task2.

The Direct Output prompt in Task2

Task:
Generate a challenging test input that exposes the bug in the buggy code of the algorithm problem:
Algorithm Problem: {problem_description}
Buggy Code: {buggy_code}

Instructions:
- Focus on edge cases or scenarios that maximize the failure probability in faulty solutions.
- Due to the output length limit, you should generate a small-scale test input that is complete and valid.
- Output the test input directly, not code to generate it.

Output format:

```plaintext
{test input}
```

Only output the test input, no explanations.

Figure 8: The Direct Output prompt used in Task2.

Are Optimal Algorithms Still Optimal? Rethinking Sorting in LLM-Based Pairwise Ranking with Batching and Caching

Juan Wisznia[1,3], Cecilia Bolaños[1,2],
Juan Tollo[1], Giovanni Marraffini[1,3], Agustín Gianolini[1],
Noe Hsueh[1], Luciano Del Corro[1,3]

{jwisznia, cbolanos, jtollo, agianolini, nhsueh, ldelcorro}@dc.uba.ar, giovanni.marraffini@gmail.com

[1]Departamento de Computación, FCEyN, Universidad de Buenos Aires
[2]Instituto de Ciencias de la Computación, FCEyN, Universidad de Buenos Aires
[3]Lumina Labs*

Abstract

We introduce a novel framework for analyzing sorting algorithms in pairwise ranking prompting (PRP), re-centering the cost model around LLM inferences rather than traditional pairwise comparisons. While classical metrics based on comparison counts have traditionally been used to gauge efficiency, our analysis reveals that expensive LLM inferences overturn these predictions; accordingly, our framework encourages strategies such as batching and caching to mitigate inference costs. We show that algorithms optimal in the classical setting can lose efficiency when LLM inferences dominate the cost under certain optimizations.

1 Introduction

LLMs have ushered in a new era of language understanding (Brown et al., 2020). Alongside these developments, LLM-based reranking has emerged in the information retrieval (IR) domain (Nogueira et al., 2020; Zhuang et al., 2023; Ma et al., 2023; Sun et al., 2024). Instead of using custom fine-tuned rankers, off-the-shelf LLMs—often combined with a first-stage retriever can refine search results in a zero-shot manner. The practical significance of reranking is evident in its rapid commercial adoption, with major cloud platforms now offering it as a core functionality. LLM-based reranking enables robust ranking quality without the overhead of dataset-specific models, which is crucial, for example, for the widespread adoption of Retrieval-Augmented Generation across both cloud-based and on-prem deployments.

A notable exemplar in zero-shot LLM-based reranking is Pairwise Ranking Prompting (PRP) (Qin et al., 2024; Luo et al., 2024), which compares two candidate documents. Despite its conceptual elegance and model-agnostic nature, PRP faces significant computational challenges; in practice—each pairwise comparison requires an expensive LLM inference, making a naive all-pairs approach prohibitively costly (Qin et al., 2024). This has prompted both researchers and practitioners to adopt classical sorting algorithms for minimizing the number of comparisons (Qin et al., 2024) as they offer theoretical guarantees.

We argue that classical analysis is not adequate for PRP as it treats each comparison as an atomic, uniform-cost operation, whereas in an LLM-based system, each comparison is an expensive inference call. This gap between classical and LLM-centric views can invert conventional wisdom under certain basic optimizations, causing algorithms that appear optimal under traditional assumptions to underperform in real-world scenarios, and vice versa.

To address these limitations, we introduce a framework that redefines how ranking algorithms are analyzed in an LLM context. Rather than merely counting comparisons, we focus on LLM inference calls as the primary cost driver. We show that basic optimizations—such as caching and batch inference—can significantly alter algorithms' efficiency. Furthermore, we propose Quicksort as an efficient reranking algorithm, demonstrating its potential when leveraging these optimizations. To the best of our knowledge, this is the first time Quicksort has been applied in this context.

Caching and Batching have no effect on algorithm ranking performance; the exact same comparisons will be performed but much faster. While caching repeated queries and batching independent operations are seemingly trivial adaptations, they significantly affect the choice of the optimal algorithm challenging previous results (Qin et al., 2024; Zhuang et al., 2024). For instance, Heapsort is no longer the preferred choice. A mere batch size of 2 will result in Quicksort generating 44% less inference calls compared to Heapsort.

We validate our findings on standard ranking benchmarks (TREC DL 2019 and 2020 (Craswell

*Work initiated at Lumina Labs.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1064–1072
July 27 - August 1, 2025

et al., 2020, 2021) and BEIR (Thakur et al., 2021)). By re-framing sorting theory around real-world LLM inference costs, we offer both practical guidance for zero-shot reranking and a theoretical basis for understanding algorithmic efficiency under modern IR constraints.

2 Related Work

Traditional IR systems require extensive labeled data and struggle with cross-domain generalization (Matveeva et al., 2006; Wang et al., 2011). LLMs have transformed this landscape by enabling zero-shot ranking. PRP emerged then as a particularly effective technique (Qin et al., 2024; Luo et al., 2024). PRP's key advantage lies in its model-agnostic nature- by comparing document pairs through simple prompts, it can leverage any LLM without training or access to model internals, making it especially valuable as newer models emerge. However, PRP faces significant computational challenges as each pairwise comparison requires an expensive LLM inference, with costs scaling quadratically with document count.

To address these computational demands, recent work has incorporated sorting algorithms into the PRP framework (Qin et al., 2024; Zhuang et al., 2024). While theoretically well-grounded, these approaches adopt the cost framework of traditional sorting theory, where comparisons are treated as atomic operations with uniform costs. However, in LLM-based ranking, inferences are orders of magnitude more expensive than other operations. This mismatch between classical cost assumptions and LLM-specific characteristics suggests the need to reevaluate sorting algorithm selection and optimization for real-world performance.

3 Revisiting sorting algorithms

This section examines how small yet impactful opimizations (caching, batching, and top-k extraction) in the context of classical algorithms—Bubblesort, Quicksort, and Heapsort can significantly shift which algorithm is most efficient in LLM-based ranking. While these adaptations are not exhaustive, they demonstrate how our framework redefines efficiency based on LLM-specific costs, where reducing inference steps matters more than traditional complexity metrics. Importantly, these optimizations preserve the final ranking outcome: the same comparisons are performed but are batched or reused, leading to fewer inference calls and a much faster process. Table 1 summarizes the optimizations applicable to each algorithm.

Algorithm	Batching	Caching	Top-k Efficiency
Heapsort	✗	✗	✓
Bubblesort	✗	✓	✓
Quicksort	✓	✗	✓[1]

Table 1: Summary of optimization techniques under LLM-centric costs.

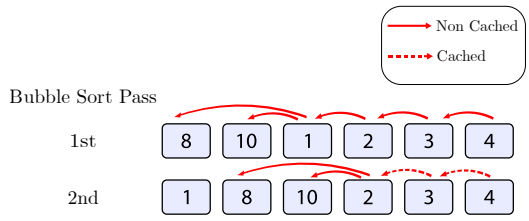


Figure 1: Bubblesort with Caching. Solid arrows show inferences, dashed arrows cached comparisons.

Heapsort has been favored in early PRP research Qin et al. (2024) for its $O(n \log n)$ complexity and natural support for top-k extraction. However, it cannot be adapted to batching or caching due to its binary tree structure. Each comparison is inherently sequential and unique. This makes it impossible to group comparisons into a single inference step (batching) or to reuse prior results (caching) effectively.

Bubblesort has been considered expensive due to its $O(n^2)$ complexity, but this can be adapted via caching from its repeated adjacent comparisons across passes (Figure 1). The memory overhead remains negligible, requiring only a small dictionary to store prior results. While its pairwise swap structure precludes batching (comparisons cannot be grouped into single inferences), it inherently supports top-k extraction (Qin et al., 2024), enhancing its practicality for ranking applications.

Quicksort uniquely enables batching through its partition phase, where multiple elements can be evaluated simultaneously against a pivot (Figure 2). However, it has limited potential for caching, as pivot comparisons are typically non-repeating. Despite this, the Partial Quicksort variant (Martınez, 2004) enhances its efficiency by enabling early termination for top-k extraction. To the best of our knowledge, we are the first to introduce Quicksort in PPR as prior research focused on Heapsort and Bubblesort due to their top-k properties.

[1] Using Partial Quicksort (Martınez, 2004).

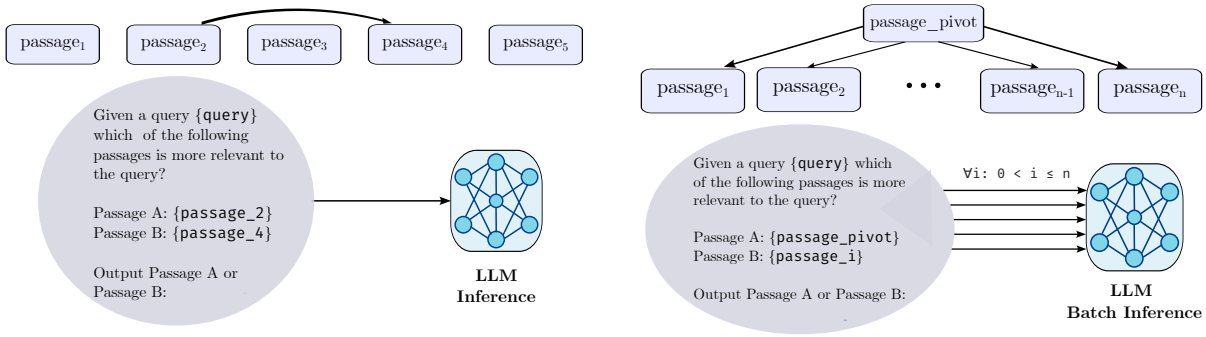


Figure 2: a. One comparison per inference as classic analysis for these algorithms. b. Multiple comparisons per inference making gaining more information per inference.

4 Experimental Setup

Hardware: Our analysis is theoretical and agnostic to hardware. However, to validate that our cost assumptions align with practical throughput behavior, we ran two lightweight empirical checks on NVIDIA A100 (40GB), RTX 3090, and RTX 2080 Ti. These include single forward-pass latency measurements across batch sizes and GPUs, and full PRP reranking with Quicksort and Heapsort at batch sizes 2 and 128 for the A100 (see Section 5).

Metric: Instead of focusing on traditional comparison counts, we shifted to the number of LLM inference calls, which are the dominant computational cost. Each inference—regardless of token count or monetary cost—is treated as a uniform cost unit. We disregard token counts and dollar costs because these are determined by the dataset and pre-trained model. Moreover, standard preprocessing (e.g., chunking/truncation) ensures uniformity across documents. We show mean and standard deviation across datasets and LLMs. Individual results can be found in Appendix A.

LLMs: Following Qin et al. (2024); Zhuang et al. (2024) we used: Flan-T5-L (780M), Flan-T5-XL (3B), Flan-T5-XXL (11B) (Chung et al., 2022), Mistral-Instruct (7B) (Jiang et al., 2023), and Llama-3-Instruct (8B) (et al, 2024). For the latency analysis we implemented batch processing with Flan-T5-Large using the Hugging Face Transformers library (Wolf et al., 2020).

Algorithms: (1) Bubblesort, (2) Quicksort with median-of-three pivot strategy (other strategies are shown in Appendix A), and (3) Heapsort.

Datasets: TREC DL 2019 (43 queries) and 2020 (200) (Craswell et al., 2020, 2021) as well as subsets from BEIR (Thakur et al., 2021): Webis-Touche2020 (49), NFCorpus (295), Large-Scifact (300), TREC-COVID (50), FiQA (648), and DBpedia-Entity (400). Following standard practices, we re-ranked the top 100 BM25-retrieved documents per query (Robertson and Zaragoza, 2009; Qin et al., 2024; Zhuang et al., 2024; Luo et al., 2024) to identify the top-10 most relevant ones efficiently.

5 Results and Discussion

Cost model analysis: Figure 3 illustrates the number of inferences performed by Heapsort and Quicksort across different batch sizes. When the batch size is set to 1 (equivalent to counting individual comparisons), Heapsort emerges as the most efficient algorithm consistent with traditional sorting analysis and previous results (Qin et al., 2024; Zhuang et al., 2024). However, as the batch size increases, Quicksort is able to significantly outperform as multiple comparisons can be run in parallel. For instance, with a batch size of 2, the average number of inference calls is reduced already by almost 45%.

Figure 4 compares the number of inferences performed by Bubblesort with and without cache. Bubblesort benefits significantly more from caching at a minimal storage overhead. This is because Bubblesort involves repeated comparisons, many of which can be cached, reducing the total number of inferences by an average of 46%.

Importantly, despite these optimizations reducing the number of LLM inferences, they do not alter the final ranking outcome. The same comparisons are performed, but they are either batched together or retrieved from cache rather than recomputed, leading to fewer inference calls and a much faster process.

Latency Analysis: Figure 5 shows single-pass

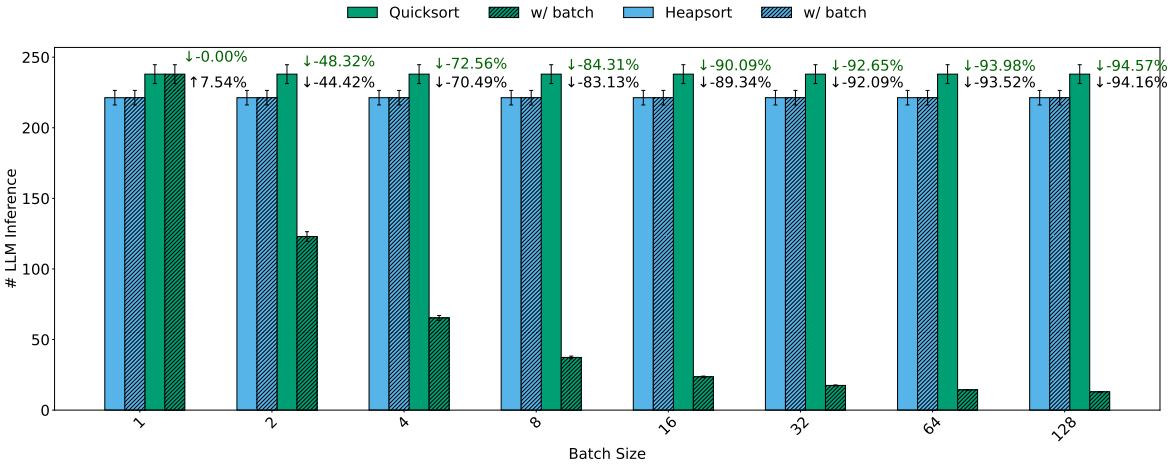


Figure 3: Mean and SD inference count for Quicksort and Heapsort across batch sizes. Black number: Heapsort vs. Quicksort using batching gain; Green number: Quicksort batching vs. no batching gain.

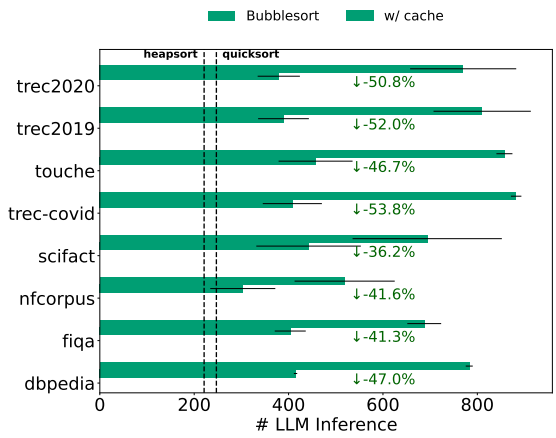


Figure 4: Mean and SD inference count for Bubblesort, with and without cache. Green numbers indicate the percentage gain with cache. The dashed line represents the mean inference count for Heapsort and Quicksort.

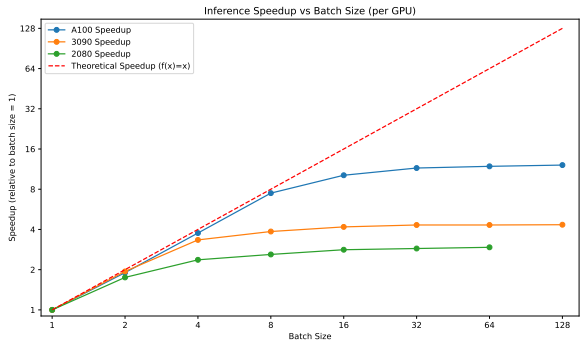


Figure 5: Speed-up vs batch size for Flan-T5-Large (log–log). Dashed red: ideal linear scaling.

speed-ups on A100, RTX 3090, and RTX 2080 Ti for different batch sizes. A100 achieves near-ideal scaling up to batch size 8, with throughput continuing to improve—albeit with diminishing returns—up to batch 128. On 3090 and 2080 Ti, ideal scaling occurs up to batch sizes 2 and 4, respectively, with throughput saturating between batch sizes 32 and 64. These results indicate that while theoretical efficiency peaks at larger batch sizes, practical efficiency is constrained by GPU architecture. The point at which near-ideal conditions are met before saturation sets in is GPU-dependent.

We also ran the full PRP pipeline over BEIR on the A100 using both batch size 2 and 128 for Quicksort and Heapsort. At batch 128, Quicksort is $5.52\times$ faster than Heapsort while achieving similar nDCG@10 (See Appendix A, Tables 2–3) . Experiments show that the theoretical gains from batching and algorithmic design hold in end-to-end ranking performance.

Ranking Performance: Figure 6 shows that the ranking performance of all these algorithms across optimization settings remains relatively stable for a given dataset, allowing users to prioritize computational efficiency and hardware constraints before performance when choosing an algorithm.

Findings provide a detailed insight of sorting algorithms behavior in LLM-based pairwise ranking, highlighting their respective benefits and drawbacks, enabling users to select the most suitable algorithm based on their specific resources and requirements. More specifically:

Quicksort is ideal for latency-sensitive applications with batch sizes ≥ 2, leveraging hardware parallelism to outperform alternatives.

Bubblesort achieves a susbtantial efficiency gain with caching. It's remarkable performance in some datasets like scifact and touche2020 makes it a more competitive choice with the new adaptation. Bubblesort tends to be effective in the context of LLMs in which pairwise transitivity is not guarantied. Pairwise adjacent comparisons seemed to be more stable and bring better results in the context of PRP (Luo et al., 2024).

Heapsort, once the gold standard for its theoretical logarithmic complexity, its advantage emerges only with no batching (rarely seen in LLM ranking).

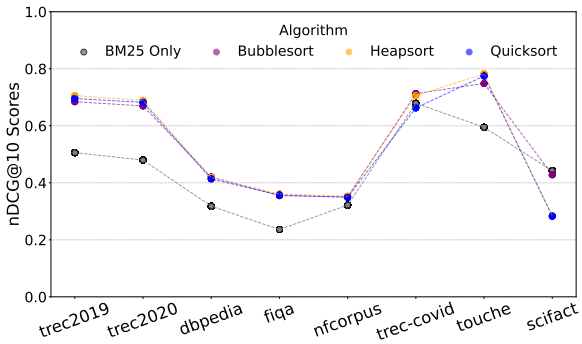


Figure 6: Algorithms' performance across datasets.

6 Conclusion

We introduced a framework for optimizing sorting algorithms in LLM-based pairwise ranking by prioritizing inference calls over comparison counts. We found that classical efficiency assumptions break down under LLM workloads, revealing Quicksort as a natural, yet unexplored, choice of algorithm. This demonstrates that inference efficiency is a property deeply tied to algorithmic design. We hope this framework encourages further exploration of algorithms better aligned with LLM cost structures.

7 Limitations

While our work showcases the efficacy of batching and caching optimizations in mitigating the high inference costs of LLM-based pairwise ranking, certain limitations remain. First, sorting algorithms work best when transitivity in pairwise comparisons holds, but LLMs can yield inconsistent judgments for near-equivalent or context-sensitive documents. Addressing this inconsistency requires dedicated methods to detect and resolve intransitive preferences, which remains an open area of research. Future work could examine how much performance is degraded and whether ranking algorithms that do not assume transitivity can actually offer any practical advantage.

Additionally, although our experiments were limited to medium-sized LLMs for budgetary and computational reasons, larger models could further amplify the benefits observed here. Future research should explore how our framework performs with these more powerful models, potentially unlocking even greater gains in inference efficiency. Moreover, hybrid methods that unify the strengths of multiple algorithms, as well as active ranking strategies or noisy sorting algorithms (Mikhailiuk et al., 2020; Bai and Coester, 2023), are fully compatible with our approach: they rely on additional computations separate from the LLM inferences themselves, thereby enabling more informed—and thus fewer—LLM queries. Ultimately, our findings underscore the need for ongoing algorithmic innovation that exploits LLM-specific cost structures, paving the way for more efficient, scalable, and broadly applicable ranking solutions.

References

Xingjian Bai and Christian Coester. 2023. Sorting with predictions.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the trec 2020 deep learning track.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the trec 2019 deep learning track.

Abhimanyu Dubey et al. 2024. The llama 3 herd of models.

C.A.R. Hoare. 1962. Quicksort. *BCS, Computer Journal*, 5(1):10–15.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Jian Luo, Xuanang Chen, Ben He, and Le Sun. 2024. PRP-graph: Pairwise ranking prompting to LLMs with graph aggregation for effective text re-ranking. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 5766–5776, Bangkok, Thailand. Association for Computational Linguistics.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model.

Conrado Martınez. 2004. Partial quicksort. In *In Proceedings of the 6th ACMSIAM Workshop on Algorithm Engineering and Experiments and 1st ACMSIAM Workshop on Analytic Algorithmics and Combinatorics*, pages 224–228.

Irina Matveeva, Chris Burges, Timo Burkard, Andy Laucius, and Leon Wong. 2006. High accuracy retrieval with multiple nested ranker. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 437–444, New York, NY, USA. Association for Computing Machinery.

Aliaksei Mikhailiuk, Clifford Wilmot, María Pérez-Ortiz, Dingcheng Yue, and Rafal Mantiuk. 2020. Active sampling for pairwise comparisons via approximate message passing and information gain maximization. *CoRR*, abs/2004.05691.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518, Mexico City, Mexico. Association for Computational Linguistics.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3:333–389.

Robert Sedgewick. 1975. *Quicksort*. Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2024. Is chatgpt good at search? investigating large language models as re-ranking agents.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models.

Lidan Wang, Jimmy Lin, and Donald Metzler. 2011. A cascade ranking model for efficient ranked retrieval. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, page 105–114, New York, NY, USA. Association for Computing Machinery.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023. Rankt5: Fine-tuning t5 for text ranking with ranking losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2308–2313, New York, NY, USA. Association for Computing Machinery.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2024. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2024, page 38–47. Association for Computing Machinery.

A Appendix

In this appendix, we present a comparison of different methods across the **BEIR** and **TREC** datasets. Each table follows the same structure and reports NDCG@10 (Normalized Discounted Cumulative Gain), the number of inferences, and the number of comparisons (#Inferences and #Comparisons) for each method. Also, latency (in seconds) is reported for Quicksort and Heapsort over the BEIR suite and corresponds to end-to-end PRP execution over each dataset on an A100 40GB GPU. Quicksort is analyzed using four pivot selection strategies: the original Hoare's method, the middle-element selection, random and median-of-three strategies (Hoare, 1962; Sedgewick, 1975).

To highlight performance differences, we emphasize the best-performing algorithm for each dataset and LLM model in **black**, while the second-best is underlined. Additionally, the tables distinguish between two computational scenarios: (1) cached: The number of inferences made to the LLM. (2) non-cached: The number of comparisons performed using precomputed results, avoiding additional inferences. All results are presented with a batch size of 2 and 128 to show batch inference efficiency.

		dbpedia				nfcorpus				fiqa			
#	Methods	NDCG@10	#comp	#inf	Lat.	NDCG@10	#comp	#inf	Lat.	NDCG@10	#comp	#inf	Lat.
	BM25	0.318	–	–	–	0.322	–	–	–	0.240	–	–	–
Flan-t5-large	heapsort #	0.413	225.1	225.1	24.8	0.335	160.0	160.0	18.4	**0.313**	200.3	200.3	26.1
	quicksort (original, b=2)	0.403	245.3	126.8	34.1	0.321	171.0	88.9	16.8	0.284	253.3	130.8	27.7
	quicksort (original, b=128)	0.403	245.3	13.2	4.7	0.321	171.0	11.1	3.0	0.284	253.3	13.6	4.8
	quicksort (random, b=2)	0.414	236.9	122.4	–	0.322	181.3	94.1	–	0.282	246.0	127.0	–
	quicksort (random, b=128)	0.405	241.5	13.0	–	0.322	171.6	10.9	–	0.289	246.6	13.2	–
	quicksort (middle, b=2)	0.410	231.9	119.9	–	0.315	168.9	87.8	–	0.277	243.2	125.6	–
	quicksort (middle, b=128)	0.410	231.9	12.8	–	0.315	168.9	10.9	–	0.277	243.2	13.2	–
	quicksort (median of three, b=2)	0.414	255.2	115.6	–	0.326	187.0	83.7	–	0.295	284.9	128.7	–
	quicksort (median of three, b=128)	0.414	255.2	12.3	–	0.326	187.0	10.5	–	0.295	284.9	13.1	–
	bubblesort (classic)	0.415	777.6	777.6	–	0.343	593.9	593.9	–	0.295	662.1	662.1	–
	bubblesort (cached)	**0.415**	777.6	360.4	–	**0.343**	593.9	242.2	–	0.295	662.1	235.3	–
Flan-t5-xl	heapsort	0.419	229.3	229.3	–	0.353	144.9	144.9	–	**0.361**	224.5	224.5	–
	quicksort (original, b=2)	0.404	238.6	123.3	–	0.345	160.6	83.7	–	0.338	209.1	108.3	–
	quicksort (original, b=128)	0.404	238.6	12.7	–	0.345	160.6	10.9	–	0.338	209.1	11.9	–
	quicksort (random, b=2)	0.412	221.6	114.7	–	0.343	168.7	87.6	–	0.345	211.3	109.5	–
	quicksort (random, b=128)	0.411	230.8	12.4	–	0.344	159.0	10.5	–	0.338	211.3	12.0	–
	quicksort (middle, b=2)	0.410	221.3	114.5	–	0.353	157.6	82.1	–	0.341	205.4	106.5	–
	quicksort (middle, b=128)	0.410	221.3	12.2	–	**0.353**	157.6	10.4	–	0.341	205.4	11.9	–
	quicksort (median of three, b=2)	0.413	234.6	106.3	–	0.349	184.9	82.8	–	0.357	226.6	102.7	–
	quicksort (median of three, b=128)	0.413	234.6	11.6	–	0.349	184.9	10.3	–	0.357	226.6	11.1	–
	bubblesort (classic)	0.420	788.2	788.2	–	0.351	443.8	443.8	–	0.355	712.8	712.8	–
	bubblesort (cached)	**0.420**	788.1	376.0	–	0.351	443.8	189.6	–	0.355	712.8	332.5	–

Table 2: Comparison of different methods across **DBPedia**, **NFCorpus**, and **FiQA** datasets.

	#	Methods	scifact				trec-covid				touche2020			
			NDCG@10	#comp	#inf	Lat.	NDCG@10	#comp	#inf	Lat.	NDCG@10	#comp	#inf	Lat.
		BM25	0.679	–	–	–	0.595	–	–	–	0.442	–	–	–
Flan-t5-large		heapsort	0.675	222.4	222.4	26.9	0.753	241.0	241.0	28.0	0.332	221.0	221.0	26.0
		quicksort (original, b=2)	0.579	211.4	109.5	25.3	0.752	245.3	126.9	26.9	0.268	273.6	141.0	34.9
		quicksort (original, b=128)	0.579	211.4	12.0	4.3	0.752	245.3	13.6	5.2	0.268	273.6	13.9	5.6
		quicksort (random, b=2)	0.596	224.7	116.2	–	0.759	243.8	126.0	–	0.270	275.2	142.0	–
		quicksort (random, b=128)	0.611	218.5	12.3	–	0.755	243.7	13.6	–	0.256	273.0	13.4	–
		quicksort (middle, b=2)	0.597	211.2	109.4	–	0.763	235.5	121.8	–	0.269	253.2	130.8	–
		quicksort (middle, b=128)	0.597	211.2	11.8	–	**0.763**	235.5	13.0	–	0.269	253.2	13.4	–
		quicksort (median of three, b=2)	0.637	237.1	107.6	–	0.763	256.0	115.1	–	0.274	289.4	131.6	–
		quicksort (median of three, b=128)	0.637	237.1	11.4	–	0.763	256.0	12.9	–	0.274	289.4	13.0	–
		bubblesort (classic)	0.692	805.7	805.7	–	0.718	890.1	890.1	–	0.447	845.4	845.4	–
		bubblesort (cached)	**0.692**	805.7	284.9	–	0.718	890.1	437.9	–	**0.447**	845.4	332.8	–
Flan-t5-xl		heapsort	0.710	197.5	197.5	–	**0.783**	249.5	249.5	–	0.284	244.3	244.3	–
		quicksort (original, b=2)	0.634	206.9	107.2	–	0.761	225.4	116.6	–	0.261	234.7	121.3	–
		quicksort (original, b=128)	0.634	206.9	11.6	–	0.761	225.4	12.5	–	0.261	234.7	12.6	–
		quicksort (random, b=2)	0.646	219.0	113.3	–	0.777	243.4	125.6	–	0.285	216.3	112.1	–
		quicksort (random, b=128)	0.639	211.8	12.0	–	0.772	222.4	12.8	–	0.265	236.7	12.9	–
		quicksort (middle, b=2)	0.642	200.4	103.9	–	0.777	239.4	123.6	–	0.277	214.7	111.0	–
		quicksort (middle, b=128)	0.642	200.4	11.5	–	0.777	239.4	12.4	–	0.277	214.7	11.9	–
		quicksort (median of three, b=2)	0.663	235.1	106.7	–	0.775	250.3	113.3	–	0.283	232.0	105.1	–
		quicksort (median of three, b=128)	0.663	235.1	11.4	–	0.775	250.3	12.4	–	0.283	232.0	11.4	–
		bubblesort (classic)	0.713	581.9	581.9	–	0.748	874.5	874.5	–	0.428	869.4	869.4	–
		bubblesort (cached)	**0.713**	581.9	217.9	–	0.748	874.5	510.9	–	**0.428**	869.4	467.7	–

Table 3: Comparison of different methods across **SciFact**, **TREC-COVID**, and **Touche2020** datasets.

#	Methods	TREC DL 2019			TREC DL 2020		
		NDCG@10	#Comparisons	#Inferences	NDCG@10	#Comparisons	#Inferences
	BM25	0.510	–	–	0.479	–	–
Flan-t5-large	heapsort	0.650	230.9	230.9	**0.626**	226.5	226.5
	quicksort (original, b=2)	0.637	249.0	128.8	0.588	237.1	122.7
	quicksort (original, b=128)	0.637	249.0	14.1	0.588	237.1	13.5
	quicksort (random, b=2)	0.639	236.7	122.3	0.587	236.7	122.4
	quicksort (random, b=128)	0.650	260.5	13.7	0.580	240.0	12.8
	quicksort (middle, b=2)	0.650	231.1	119.6	0.594	235.5	121.8
	quicksort (middle, b=128)	0.650	231.1	13.5	0.594	235.5	13.0
	quicksort (median of three, b=2)	0.650	276.0	124.8	0.600	259.5	117.0
	quicksort (median of three, b=128)	**0.650**	276.0	13.2	0.600	259.5	12.8
	bubblesort (classic)	0.634	843.7	843.7	0.586	777.2	777.2
	bubblesort (cached)	0.634	843.7	388.3	0.586	777.2	357.1
Flan-t5-xl	heapsort	**0.706**	242.0	242.0	**0.689**	244.9	244.9
	quicksort (original, b=2)	0.697	266.6	137.5	0.672	250.6	129.3
	quicksort (original, b=128)	0.697	266.6	13.9	0.672	250.6	12.9
	quicksort (random, b=2)	0.694	232.3	120.2	0.676	239.0	123.5
	quicksort (random, b=128)	0.697	257.3	13.2	0.676	237.6	12.7
	quicksort (middle, b=2)	0.703	230.5	119.4	0.668	232.9	120.5
	quicksort (middle, b=128)	0.703	230.5	13.1	0.668	232.9	12.6
	quicksort (median of three, b=2)	0.696	243.5	110.5	0.682	239.6	108.9
	quicksort (median of three, b=128)	0.696	243.5	11.9	0.682	239.6	11.8
	bubblesort (classic)	0.684	887.8	887.8	0.670	869.5	869.5
	bubblesort (cached)	0.684	887.8	544.9	0.670	869.5	542.6
Flan-t5-xxl	heapsort	**0.702**	238.9	238.9	0.688	239.4	239.4
	quicksort (original, b=2)	0.677	265.9	137.0	0.680	234.7	121.3
	quicksort (original, b=128)	0.677	265.9	13.6	0.680	234.7	12.7
	quicksort (random, b=2)	0.691	239.4	124.0	0.678	228.3	117.9
	quicksort (random, b=128)	0.685	244.7	12.8	0.674	227.4	12.3
	quicksort (middle, b=2)	0.688	226.3	117.0	0.677	229.2	118.5
	quicksort (middle, b=128)	0.688	226.3	12.3	0.677	229.2	12.3
	quicksort (median of three, b=2)	0.686	254.8	116.3	0.688	230.4	104.7
	quicksort (median of three, b=128)	0.686	254.8	11.9	**0.688**	230.4	11.4
	bubblesort (classic)	0.679	866.2	866.2	0.680	827.1	827.1
	bubblesort (cached)	0.679	866.2	532.1	0.680	827.1	465.0
Meta-Llama-3-8B-Instruct	heapsort	0.662	235.0	235.0	**0.615**	231.9	231.9
	quicksort (original, b=2)	0.645	266.5	137.4	0.576	235.5	121.8
	quicksort (original, b=128)	0.645	266.5	13.5	0.576	235.5	12.9
	quicksort (random, b=2)	**0.663**	231.3	119.8	0.580	231.7	119.8
	quicksort (random, b=128)	0.660	219.0	12.8	0.585	232.9	12.8
	quicksort (middle, b=2)	0.640	220.9	114.4	0.564	228.1	118.0
	quicksort (middle, b=128)	0.640	220.9	12.6	0.564	228.1	12.6
	quicksort (median of three, b=2)	0.650	236.0	106.3	0.594	244.3	110.1
	quicksort (median of three, b=128)	0.650	236.0	12.0	0.594	244.3	12.3
	bubblesort (classic)	0.641	822.5	822.5	0.600	797.6	797.6
	bubblesort (cached)	0.641	822.5	389.4	0.600	797.6	365.9
Mistral-7B-Instruct-v0.1	heapsort	0.559	200.3	200.3	0.513	190.1	190.1
	quicksort (original, b=2)	0.578	278.9	143.7	0.529	276.2	142.2
	quicksort (original, b=128)	0.578	278.9	14.0	0.529	276.2	13.4
	quicksort (random, b=2)	0.593	293.8	150.9	0.511	271.5	139.8
	quicksort (random, b=128)	0.573	279.7	12.9	0.524	263.9	13.2
	quicksort (middle, b=2)	0.595	257.3	132.7	0.531	249.3	128.6
	quicksort (middle, b=128)	0.595	257.3	13.5	0.531	249.3	12.9
	quicksort (median of three, b=2)	0.612	292.1	132.5	0.538	292.5	133.0
	quicksort (median of three, b=128)	**0.612**	292.1	13.1	0.538	292.5	12.9
	bubblesort (classic)	0.587	631.0	631.0	0.539	578.7	578.7
	bubblesort (cached)	0.587	631.0	250.5	**0.539**	578.7	223.4

Table 4: Comparison of methods for **TREC DL 2019** and **TREC DL 2020**.

TreeCut: A Synthetic Unanswerable Math Word Problem Dataset for LLM Hallucination Evaluation

Jialin Ouyang
Columbia University
jo2559@columbia.edu

Abstract

Large language models (LLMs) now achieve near-human performance on standard math word problem benchmarks (e.g., GSM8K), yet their true reasoning ability remains disputed. A key concern is that models often produce confident, yet unfounded, answers to unanswerable problems. We introduce TreeCut, a synthetic dataset that systematically generates *infinite* unanswerable math word problems and their answerable counterparts, by representing each question as a tree and removing chosen necessary conditions. Experiments show TreeCut effectively induce hallucinations in large language models, including GPT-4o and o3-mini, with rates of 64% and 44% in their respective worst-case scenarios under zero-shot setting. Further analysis highlights that deeper or more complex trees, composite item names, and removing necessary condition near the middle of a path all increase the likelihood of hallucinations, underscoring the persistent challenges LLMs face in identifying unanswerable math problems. The dataset generation code and sample data are available at https://github.com/j-bagel/treecut-math.

1 Introduction

Mathematical reasoning is a crucial part of human intelligence. Recent years have witnessed remarkable advancements in the mathematical reasoning capabilities of large language models (LLMs). By leveraging techniques such as chain-of-thought prompting (Wei et al., 2022), state-of-the-art LLMs (e.g., Achiam et al. (2023); Team et al. (2024); Dubey et al. (2024)) achieved human-level performance on benchmarks like GSM8K (Cobbe et al., 2021). However, it remains controversial whether this performance implies reasoning capability beyond pattern matching.

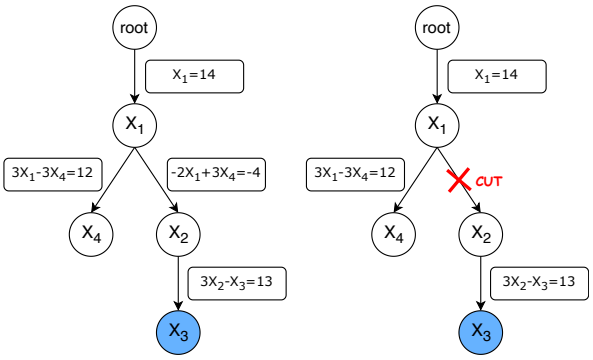
A substantial body of research highlights the capability of Large Language Models in mathematical reasoning. Achiam et al. (2023); Team et al. (2024); Dubey et al. (2024); Yang et al. (2024), among others, achieved over 90% accuracy on GSM8K (Cobbe et al., 2021), a dataset consists of 8K grade school math word problems. Yang et al. (2024); Zhou et al. (2023), among others, achieved over 80% accuracy on the more difficult MATH dataset (Hendrycks et al., 2021), which consists of 12.5K high school math competition problems.

Meanwhile, there is a line of research questioning the reasoning ability of LLMs by showing their vulnerability under superficial changes of the input that do not alter the underlying logic. Works like Shi et al. (2023); Jiang et al. (2024) find that LLMs are easily distracted by irrelevant context or token level perturbation that does not change the underlying logic of the reasoning task. Mirzadeh et al. (2024) further demonstrate that the performance of LLMs declines when numerical values are altered in the questions from the GSM8K dataset.

There is yet another line of research that challenges the ability of LLMs to refrain from answering unanswerable problems. Ma et al. (2024); Li et al. (2024); Sun et al. (2024); Zhou et al. (2024a); Saadat et al. (2024) introduce minor modifications to existing math word problems to create unanswerable variants, and find that LLMs often generate hallucinatory answers for these unanswerable questions, even when they perform well on the original answerable datasets. However, these efforts rely on pre-existing math word problem sources, making them susceptible to training data contamination, limited in scope, and lacking rich structures for extended research.

To address these shortcomings, we propose TreeCut, a synthetic dataset capable of systematically generating an infinite number of unanswerable math word problems and their answerable counterparts. TreeCut considers problem represented by a tree, with nodes representing variables and edges representing formulas. Unanswerable problems are generated by removing an edge along

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1073–1085
July 27 - August 1, 2025



Question:
A burger costs 14 dollars. ~~3 scrambled eggs cost 4 dollars less than 2 burgers.~~ 3 pies cost 12 dollars less than 3 burgers. A BLT sandwich costs 13 dollars less than 3 scrambled eggs. Question: how much does a BLT sandwich cost?

Solution to the answerable problem:
It is given as a fact that a burger costs 14 dollars. Combine with the fact that 3 scrambled eggs cost 4 dollars less than 2 burgers, we get a scrambled egg costs 8 dollars. Combine with the fact that a BLT sandwich costs 13 dollars less than 3 scrambled eggs, we get a BLT sandwich costs 11 dollars.

Solution to the unanswerable problem:
All we know about the prices of BLT sandwich and scrambled egg is: a BLT sandwich costs 13 dollars less than 3 scrambled eggs. There are 2 variables but only 1 linear formula, so we cannot calculate the price of a BLT sandwich.

Figure 1: The left and middle panels depict the tree structures corresponding to the answerable and unanswerable questions, respectively. In the right panel, the strike-through sentence represents the formula removed by the *cut*. The variable mappings to items are as follows: x_1 represents a burger, x_2 represents a scrambled egg, x_3 represents a BLT sandwich, and x_4 represents a pie.

the path from the root to the questioned variable. Our unanswerable dataset proves to be challenging even for GPT-4o and o3-mini. In addition, `TreeCut` allows precise control over the structural components of each problem, enabling detailed investigations into when and why LLMs produce hallucinations. Our analysis highlights that deeper or more complex trees, composite item names, and removing necessary condition near the middle of a path all increase the likelihood of hallucinations.

2 Related Work

Math Word Problem Benchmark Numerous math word problem datasets of different difficulty have been proposed in previous research, most notable examples including GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021).

Many benchmarks have been developed to measure the robustness of mathematical reasoning. Patel et al. (2021); Kumar et al. (2021); Xu et al. (2022); Li et al. (2024); Zhou et al. (2024b); Yu et al. (2023); Shi et al. (2023) perturb or rewrite math word problems to measure the robustness of mathematical reasoning.

Liu et al. (2021) utilize tree structures to represent and manipulate mathematical expressions during the reverse operation based data augmentation process for MWP solving. Opedal et al. (2024) introduced MathGAP, a framework for evaluating LLMs using synthetic math word problems with controllable proof tree characteristics. In contrast to their approach, the tree structure in our problem-generation procedure is fundamentally different. In our work, each node represents a variable, and the questioned variable appears as a leaf. In their work, however, each node represents a logical statement, with the answer represented by the root. More importantly, we focus on unanswerable math word problems, an aspect that their study did not address.

Unanswerable Math Problems Yin et al. (2023) introduced SelfAware, consisting of unanswerable questions from five diverse categories. It includes less than 300 unanswerable mathematical problems. Li et al. (2024); Zhou et al. (2024a) generate unanswerable questions by prompting GPT4 to eliminate a necessary condition from the original problem, an then the modified questions are further checked or refined by human annotators. Ma et al. (2024) prompt GPT4 to modify problems from GSM8K. Sun et al. (2024) task human annotators to modify original questions in existing MWP datasets to make them unanswerable, creating a dataset composed of 2,600 answerable questions and 2,600 unanswerable questions.

3 TreeCut: a Synthetic (Un)answerable Math Word Problem Dataset

For the purpose of our investigation, we aim to have full control over the various aspects that determine the underlying *structure* of a math word problem: the name of the entities, the numeric values, and the complexity of the problem. Furthermore, we seek to reliably generate unanswerable problems by precisely removing specific necessary conditions of our choosing.

To this end, we start with a special kind of *answerable* math word problem that can be represented as a tree, as illustrated in Figure 1. Within such a tree, each non-root node represents a variable, while the root is a uniquely reserved node. An edge from root gives value to a variable, while

ansDepth	Llama-8B	Llama-70B	Qwen-7B	Qwen-72B	GPT-4o	o3-mini
2	80.2%	24.6%	84.6%	59.8%	12.0%	44.0%
4	86.2%	40.2%	90.4%	82.8%	18.0%	25.2%
6	86.0%	63.4%	95.6%	88.4%	47.4%	19.2%
8	84.2%	65.0%	93.4%	85.2%	64.0%	25.6%

Table 1: Percentage of hallucination of various LLMs at different ansDepth values for unanswerable problems, zero-shot prompting

ansDepth	Llama-8B	Llama-70B	Qwen-7B	Qwen-72B	GPT-4o	o3-mini
2	72.8%	33.6%	80.4%	55.4%	18.4%	3.2%
4	79.0%	57.6%	94.6%	84.8%	28.8%	2.4%
6	79.6%	72.4%	92.6%	84.8%	41.8%	3.6%
8	78.6%	76.8%	94.4%	83.0%	51.0%	3.0%

Table 2: Percentage of hallucination of various LLMs at different ansDepth values for unanswerable problems, few-shot prompting

an edge between two variables represents a linear formula of the two neighboring nodes. Given such a tree, any variable can be calculated following the unique path from the root to the node that represents the variable. Such a solving procedure does *not* require solving a linear equation system, as the solution only consists of carrying out basic arithmetic operations along the path. To guarantee that the arithmetic operations are well within the capacity of current frontier LLMs,we further restrict the unit price of each food item to be an integer between 5 and 15, and the coefficients of each linear equation taking non-zero integer values between -3 and 3. Finally, variables are randomly mapped to items, and then the formulas are translated to natural language using templates. The complete generation procedure, along with the templates used, is provided in Appendix A.

From an answerable math word problem described above, we generate an unanswerable problem by removing an edge along the path from the root to the questioned variable. In Figure 1, x_3 is the questioned variable. Along the path to the root, we remove the edge between x_1 and x_2 (denoted by a *cut*), rendering x_2 and x_3 undetermined, thus making the question unanswerable, as all we know about x_2 and x_3 is one single linear equation. A key benefit of such a generation procedure is that the distance from the questioned variable to the *cut* is also fully controlled, as we will see that this factor plays an important role in triggering LLM hallucination.

In summary, we can control the *structure* of problems via the following parameters:

- numVars: total number of variables,
- ansDepth: distance from the root to the questioned variable,
- compositeName: boolean, whether the items in the question have composite names (e.g. "a burger at Bistro Nice" versus "a burger"),
- cutDepth: distance from the questioned variable to the *cut*, if an unanswerable problem is to be generated.

Appendix A contains the detailed problem generation algorithm.

4 Experiments

We evaluate several state-of-the-art LLMs using TreeCut. Additionally, we analyze the hallucination rate of GPT-4o on unanswerable problems generated under different parameter configurations of TreeCut.

4.1 Experimental Setup

For each set of generation parameters, we randomly generate 500 problems. Unless stated otherwise, we employ a zero-shot prompting template that explicitly directs the model to indicate when a question is unanswerable due to insufficient conditions. A chain-of-thought system message is incorporated for all models except o3-mini[1].

4.2 Evaluating LLMs

In the first set of experiments, we generate unanswerable math word problems of varying difficulty to evaluate the following LLMs: Llama 3.1 Instruct with 8B and 70B parameters(Dubey et al., 2024), Qwen2.5 Instruct with 7B and 72B parameters(Yang et al., 2024), GPT-4o(Achiam et al., 2023), and o3-mini(OpenAI, 2025).

[1]Following OpenAI's guidelines of reasoning models.

ansDepth	Llama-8B	Llama-70B	Qwen-7B	Qwen-72B	GPT-4o	o3-mini
2	68% (14%)	95% (1%)	87% (2%)	95% (1%)	99% (1%)	100% (0%)
4	28% (12%)	82% (6%)	31% (6%)	86% (6%)	94% (0%)	100% (0%)
6	17% (16%)	83% (3%)	12% (9%)	80% (7%)	85% (3%)	100% (0%)
8	5% (12%)	76% (7%)	7% (10%)	68% (8%)	84% (2%)	100% (0%)

Table 3: Accuracy of various LLMs at different ansDepth levels for answerable problems. The percentage in parentheses represents the proportion of answerable questions incorrectly identified as unanswerable.

Table 1 summarizes the results. None of the LLMs gives satisfactory results. Llama 3.1 8B, Qwen2.5 7B and 72B barely have any success identifying unanswerable problems. Llama 3.1 70B and GPT-4o struggle with more complex problems (ansDepth $= 6, 8$). o3-mini has the lowest hallucination for ansDepth $= 6, 8$. However, for the easiest case where ansDepth $= 2$ (in this setting, only 4 variables are mentioned in each problem), o3-mini displays a bias of making hallucinatory assumptions (see Appendix C.2 for examples).

To further investigate whether the LLMs face intrinsic challenges in recognizing unanswerable math word problems, we conduct another set of experiments using few-shot prompting. For each unanswerable problem, we construct a few-shot prompt by randomly selecting 3 answerable and 3 unanswerable problems, each accompanied by a full solution path and the correct final answer. We use sample size n=500. Results are summarized in Table 2. O3-mini greatly benefits from few-shot prompting, which is not surprising given our analysis in Appendix C.2. For shorter problems, o3-mini tends to recognize the lack of conditions during reasoning, but choose to make unreasonable assumptions to arrive at a final answer. Few-shot examples guide it to refrain from doing that. The hallucination rates of the other models remained largely unchanged. This suggests that the five models besides o3-mini face intrinsic challenges in recognizing unanswerable math word problems.

To investigate whether the unsatisfactory accuracy of identifying unanswerable problems comes from the incapability of the necessary mathematical *operations*, we evaluate the LLMs on the *answerable* counterparts of the unanswerable questions using the same zero-shot prompting template. For this experiment, a sample size of $n = 100$ is used. We observe that almost every model displays a significant gap between its ability of solving answerable problems and identifying unanswerable problems. For instance, GPT-4o correctly solves 84% of answerable problems for ansDepth $= 8$, but only correctly recognizes 36% of unanswerable problems.

4.3 Unanswerable Problem Structure and Hallucination

For a more fine-grained investigation of LLM's hallucination behavior under different *structures* of unanswerable problems, we analyze GPT-4o's hallucination rate on unanswerable problems generated under different parameter choices of numVars, ansDepth, compositeName and cutDepth.

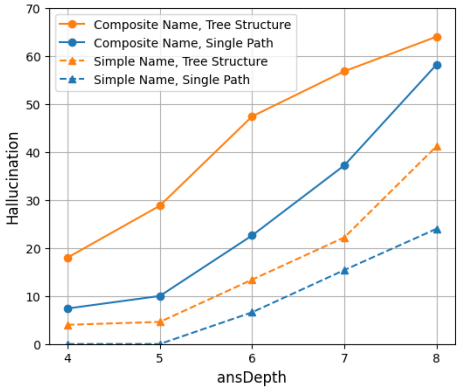


Figure 2: Hallucination percentage under different configurations of unanswerable problems, plotted against varying ansDepth.

Tree Structure and Item Names To investigate the effect of (i) a deeper tree structure, (ii) a more complex tree structure, and (iii) composite item names, we consider the following parameter configurations:

- ansDepth $\in \{4, 5, 6, 7, 8\}$, which controls the depth of the questioned variable,
- cutDepth $= \lfloor$ansDepth$/2 \rfloor$
- numVars $=$ ansDepth $+ 2$ (generates a more complex tree structure, with conditions unrelated to the questioned variable) or numVars $=$ ansDepth (the tree structure degenerates into a single path),
- compositeName: $true$ or $false$.

There are $5 \times 2 \times 2 = 20$ configurations in total. We randomly generate 500 unanswerable problems for each configuration, and summarize GPT-4o's hallucination rate in Figure 2. In the figure,

⋆ Orange line represents complex tree structure,
⋆ blue line represents simple tree structure,
- Solid line stands for composite item names,
- Dashed line stands for simple item names.

Examining each line individually, we observe that the hallucination rate increases as the depth of the questioned variable grows. Comparing solid and dashed lines of the same color, a more complex tree structure consistently results in a higher likelihood of hallucination across different `ansDepth` values. Comparing orange and blue lines of the same linestyle, composite item names consistently lead to a higher likelihood of hallucination compared to simple item names.

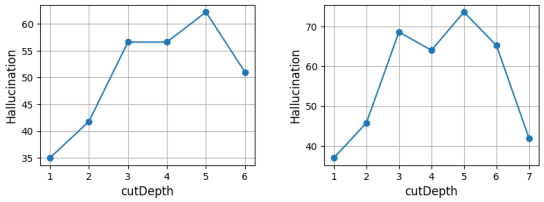


Figure 3: Hallucination percentage versus `cutDepth`. Left panel has `ansDepth` $= 7$. Right panel has `ansDepth` $= 8$.

Location of the *Cut* For each unanswerable problem, the *cut* always happens along the path from the root to the questioned variable. Does the location of the *cut* change hallucination ratio? We vary `cutDepth` from 1 to 7 while keeping `ansDepth` $=$ 8 and other parameters fixed. In the right panel of Figure 3, we see that `cutDepth` $= 3, 4, 5, 6$ all trigger over 60% hallucination for GPT-4o (with `cutDepth` $= 5$ triggering over 70%), but a `cutDepth` $= 1, 2, 7$ only triggers less than 50% of hallucination, which means that GPT-4o is more confused when the *cut* happens around the middle point along the path, comparing to that happens near the root or the questioned variable.

4.4 Conclusion of Experiments

Our findings indicate that the unanswerable math word problems generated by TREECUT effectively induce hallucinations in large language models, including GPT-4o and o3-mini, with rates of 61% and 42% in their respective worst-case scenarios. Focusing on GPT-4o, we further observe that hallucinations are more likely to occur when the problem exhibits (i) a deeper tree structure, (ii) a more complex tree structure, (iii) composite item names, or (iv) a *cut* positioned around the middle of the path. These results underscore the challenges LLMs face in handling unanswerable math problems.

Limitations

Our synthetic dataset is specifically designed for math word problems, representing only a small subset of the broader field of mathematics. Additionally, our evaluations are based on zero-shot and few-shot chain-of-thought prompting. We do not explore alternative prompting techniques commonly used in LLM-based mathematical reasoning studies, which may impact performance comparisons.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback, which helped improve the quality of this work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Bowen Jiang, Yangxinyu Xie, Zhuoqun Hao, Xiaomeng Wang, Tanwi Mallick, Weijie J Su, Camillo J Taylor, and Dan Roth. 2024. A peek into token bias: Large language models are not yet genuine reasoners. *arXiv preprint arXiv:2406.11050*.

Vivek Kumar, Rishabh Maheshwary, and Vikram Pudi. 2021. Adversarial examples for evaluating math word problem solvers. *arXiv preprint arXiv:2109.05925*.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. 2024. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. *arXiv preprint arXiv:2402.19255*.

Qianying Liu, Wenyu Guan, Sujian Li, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2021. Roda: Reverse operation based data augmentation for solving math word problems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1–11.

Jingyuan Ma, Damai Dai, Lei Sha, and Zhifang Sui. 2024. Large language models are unconscious of unreasonability in math problems. *arXiv preprint arXiv:2403.19346*.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

Andreas Opedal, Haruki Shirakami, Bernhard Schölkopf, Abulhair Saparov, and Mrinmaya Sachan. 2024. Mathgap: Out-of-distribution evaluation on problems with arbitrarily complex proofs. *arXiv preprint arXiv:2410.13502*.

OpenAI. 2025. Openai o3 mini. Accessed: Feb. 5, 2025.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094.

Asir Saadat, Tasmia Binte Sogir, Md Taukir Azam Chowdhury, and Syem Aziz. 2024. When not to answer: Evaluating prompts on gpt models for effective abstention in unanswerable math word problems. *arXiv preprint arXiv:2410.13029*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, pages 31210–31227.

YuHong Sun, Zhangyue Yin, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Hui Zhao. 2024. Benchmarking hallucination in large language models based on unanswerable math word problem. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2178–2188.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jialiang Xu, Mengyu Zhou, Xinyi He, Shi Han, and Dongmei Zhang. 2022. Towards robust numerical question answering: Diagnosing numerical capabilities of nlp systems. *arXiv preprint arXiv:2211.07455*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuan-Jing Huang. 2023. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, et al. 2023. Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification. *arXiv preprint arXiv:2308.07921*.

Zihao Zhou, Shudong Liu, Maizhen Ning, Wei Liu, Jindong Wang, Derek F Wong, Xiaowei Huang, Qiufeng Wang, and Kaizhu Huang. 2024a. Is your model really a good math reasoner? evaluating mathematical reasoning with checklist. *arXiv preprint arXiv:2407.08733*.

Zihao Zhou, Qiufeng Wang, Mingyu Jin, Jie Yao, Jianan Ye, Wei Liu, Wei Wang, Xiaowei Huang, and Kaizhu Huang. 2024b. Mathattack: Attacking large language models towards math solving ability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 19750–19758.

A Generation of the Math Word Problems

Algorithm 1 Generating Math Word Problem using Random Tree

Require: `numVars` $\geq$ `ansDepth` ≥ 2
Require: `unanswerable` $\in \{true, false\}$, `order` $\in$ {"forward", "backward", "random"}
Require: `cutDepth: int`
if `unanswerable` $= true$ **then**
Require: `cutDepth: int`, satisfying $1 \leq$ `cutDepth` $<$ `ansDepth`
end if
▷ (i) Sample a dictionary of variable values
`varDict` $\leftarrow \{\}$
for $i \leftarrow 1$ to `numVars` **do**
    Sample an integer $v \in [5, 15]$
    `varDict`$[x_i] \leftarrow v$
end for
▷ (ii) Build the random tree
Assign `root` as the parent of x_1
for $i \leftarrow 2$ to `ansDepth` **do**
    Assign x_{i-1} as the parent of x_i
end for ▷ Finish building the path from the root to the questioned variable
▷ Assign the remaining nodes
for $i \leftarrow$ `ansDepth` $+ 1$ to `numVars` **do**
    Randomly select a node x_p in the tree
    Assign x_p as the parent of x_i
end for
▷ (iii) Get the list of all edges via a breadth-first traversal
`edgeList` $\leftarrow$ the list of edges collected by a breadth-first traversal (see Algorithm 2)
▷ (iv) For unanswerable problems, create the *cut*
if `unanswerable` $= true$ **then**
    Remove $(x_{\texttt{ansDepth-cutDepth-1}}, x_{\texttt{ansDepth-cutDepth}})$ from `edgeList`
end if
▷ (v) Generate a formula for each edge, and store in `forumlaList`
`formulaList` $\leftarrow [\,]$
for edge (x_i, x_j) in `edgeList` **do**
    Sample $a, b \in \{-3, -2, -1, 1, 2, 3\}$
    Define `formula` $\leftarrow a \cdot x_i + b \cdot x_j = a \cdot$ `varDict`$[x_i] + b \cdot$ `varDict`$[x_j]$
    Append `formula` to `formulaList` ▷ So that `formulaList` has the same order as `edgeList`
end for
▷ (vi) Adjust the ordering of `formulaList` according to `order`
if `order` $=$ "backward" **then**
    Reverse `formulaList`
end if
if `order` $=$ "random" **then**
    Random Shuffle `formulaList`
end if
return `formulaList` ▷ Formulas serving as conditions of the problem.

Algorithm 1 generates `formulaList`, which contains the formulas that will serve as the conditions of the problem. To translate that into natural language, item names will be sampled according to the *compositeName* option. Then, `formulaList` can be translated to natural language using pre-defined templates. The question sentence will simply be "what is the price of `{item name of the questioned`

variable}".

We want to point out that although all the variables are assigned a value in varDict, this is purely for the sake of (i) subsequently generating the random formulas (ii) guaranteeing that all calculable variables will have values between 5 and 15. When unanswerable $= true$, the *cut* will guarantee that the problem is unanswerable.

In the following, we also detail the simple breadth-first traversal algorithm for getting all the edges from the tree, which enables us to control the order of the conditions in the problem.

Algorithm 2 Breadth-First Traversal to Get Edges

Require: root: the root of a tree

▷ Get the list of all edges via a breadth-first traversal

```
edgeList ← [ ], q ← a queue containing root
while q is not empty do
    node ← q.dequeue()
    for child ∈ node.children do
        Add (node, child) to edgeList
        Add child to q
    end for
end while
    return edgeList
```

Given a formulaList, each formula is translated via the following template:

Formula Translation Template

Suppose x_1 stands for <dish1> at <restaurant1>, and x_2 stands for <dish2> at <restaurant2>.

1. If the formula is $a_1x_1 + a_2x_2 = b$ where $a_1, a_2 > 0$, then the formula will be translated to: "a_1 <dish1> at <restaurant1> and a_2 <dish2> at <restaurant2> cost b dollars".
2. If the formula is $a_1x_1 - a_2x_2 = 0$ where $a_1, a_2 > 0$, then the formula will be translated to: "The price of a_1 <dish1> at <restaurant1> is the same as the price of a_2 <dish2> at <restaurant2>".
3. If the formula is $a_1x_1 - a_2x_2 = b$ where $a_1, a_2 > 0$ and $b > 0$, then the formula will be translated to either "a_1 <dish1> at <restaurant1> cost b dollars more than a_2 <dish2> at <restaurant2>" or "a_2 <dish2> at <restaurant2> cost b dollars less than a_1 <dish1> at <restaurant1>".
4. The case of $b < 0$ is similar to the case above.

B Details of Experiments

B.1 Prompt Template

Below is the prompt template used for evaluation, which is a 0-shot prompting template with chain-of-thought system message. For reasoning models like o3-mini, we do not use the chain-of-thought system message.

Zero-shot Evaluation Prompt Template

```
// system message (for non-reasoning models)
As an expert problem solver, solve step by step the following mathematical
questions.

// user message
Please solve the following math question, and then answer in the form 'Answer:
x'. If the known conditions are not sufficient to answer the question, please
answer in the form 'Answer: unknown.'.

Question: {{question}}

Your solution:
```

B.2 Evaluation

For all the models other than o3-mini, we set `{max_tokens=4000, temperature=0}`. o3-mini does not allow setting `temperature`, and we set `{max_completion_tokens=32000, reasoning_effort="high"}`. GPT-4o and o3-mini are evaluated via calling OpenAI API. The exact version numbers are "gpt-4o-2024-08-06" and "o3-mini-2025-01-31", respectively.

When evaluating model's answer to an unanswerable problem, we change the response to all lower case, and if "unknown" is mentioned after the last occurrence of the trigger word "answer", it is counted as a correct answer.

C Examples of Hallucinatory Answers

C.1 GPT-4o

Example 1: A BLT sandwich at Urban Plate costs 13 dollars. 3 burgers at Taste Good Cuisine cost 6 dollars more than 3 fruit tarts at Taste Good Cuisine. A BLT sandwich at Taste Good Cuisine costs 5 dollars. A fruit tart at Taste Good Cuisine costs 9 dollars less than 3 burgers at Urban Plate. 2 burgers at Taste Good Cuisine cost 8 dollars more than a fruit tart at Urban Plate. Question: how much does a burger at Taste Good Cuisine cost?

Model answer:

```
To solve this problem, we need to set up equations based on the information given and solve for the unknowns.

Let's define the variables:
- Let \( B_T \) be the cost of a burger at Taste Good Cuisine.
- Let \( F_T \) be the cost of a fruit tart at Taste Good Cuisine.
- Let \( B_U \) be the cost of a burger at Urban Plate.
- Let \( F_U \) be the cost of a fruit tart at Urban Plate.

We are given the following information:

1. A BLT sandwich at Urban Plate costs 13 dollars.
2. 3 burgers at Taste Good Cuisine cost 6 dollars more than 3 fruit tarts at Taste Good Cuisine.
   \[
   3B_T = 3F_T + 6
   \]
3. A BLT sandwich at Taste Good Cuisine costs 5 dollars.
4. A fruit tart at Taste Good Cuisine costs 9 dollars less than 3 burgers at Urban Plate.
   \[
   F_T = 3B_U - 9
   \]
5. 2 burgers at Taste Good Cuisine cost 8 dollars more than a fruit tart at Urban Plate.
   \[
   2B_T = F_U + 8
   \]
```

```
We need to find the cost of a burger at Taste Good Cuisine, \( B_T \).

From equation (2), we can express \( F_T \) in terms of \( B_T \):
\[
F_T = B_T - 2
\]

Substitute this expression for \( F_T \) into equation (4):
\[
B_T - 2 = 3B_U - 9
\]
\[
B_T = 3B_U - 7
\]

Now, let's substitute \( F_T = B_T - 2 \) into equation (5):
\[
2B_T = (B_T - 2) + 8
\]
\[
2B_T = B_T + 6
\]
\[
B_T = 6
\]

Thus, the cost of a burger at Taste Good Cuisine is 6 dollars.

Answer: 6
```

Mistake: When the model did "`Now, let's substitute F_T = B_T - 2 into equation (5)`", it actually confused `F_T` with `F_U` in `equation (5)`.

Example 2: 3 Greek salads at Texas BBQ cost 6 dollars less than 3 pieces of cheese cake at Texas BBQ. 3 burgers at Texas BBQ and 2 pieces of cheese cake at Texas BBQ cost 59 dollars. 3 burgers at Taste Good Cuisine and 2 Greek salads at Taste Good Cuisine cost 42 dollars. 2 burgers at Texas BBQ and a piece of cheese cake at Taste Good Cuisine cost 32 dollars. 3 burgers at Texas BBQ cost 15 dollars more than 3 fruit tarts at Taste Good Cuisine. A burger at Taste Good Cuisine costs 10 dollars. A fruit tart at Texas BBQ costs 2 dollars more than a Greek salad at Texas BBQ. Question: how much does a Greek salad at Texas BBQ cost?

Model answer:

```
To solve this problem, we need to set up equations based on the information given and solve for the cost of
a Greek salad at Texas BBQ. Let's define the variables:

- \( G \): Cost of a Greek salad at Texas BBQ
- \( C \): Cost of a piece of cheesecake at Texas BBQ
- \( B \): Cost of a burger at Texas BBQ
- \( F \): Cost of a fruit tart at Taste Good Cuisine
- \( B_T \): Cost of a burger at Taste Good Cuisine (given as $10)
- \( F_T \): Cost of a fruit tart at Texas BBQ

Now, let's translate the information into equations:

1. "3 Greek salads at Texas BBQ cost 6 dollars less than 3 pieces of cheesecake at Texas BBQ."
   \[
   3G = 3C - 6
   \]

2. "3 burgers at Texas BBQ and 2 pieces of cheesecake at Texas BBQ cost 59 dollars."
   \[
   3B + 2C = 59
   \]

3. "3 burgers at Taste Good Cuisine and 2 Greek salads at Taste Good Cuisine cost 42 dollars."
   \[
   3B_T + 2G = 42
   \]
```

```
   Since \( B_T = 10 \), substitute:
   \[
   3(10) + 2G = 42 \implies 30 + 2G = 42 \implies 2G = 12 \implies G = 6
   \]

4. "2 burgers at Texas BBQ and a piece of cheesecake at Taste Good Cuisine cost 32 dollars."
   \[
   2B + C_T = 32
   \]
   (Note: We don't have enough information about \( C_T \), the cost of cheesecake at Taste Good Cuisine,
   so we can't use this equation directly.)

5. "3 burgers at Texas BBQ cost 15 dollars more than 3 fruit tarts at Taste Good Cuisine."
   \[
   3B = 3F + 15
   \]

6. "A fruit tart at Texas BBQ costs 2 dollars more than a Greek salad at Texas BBQ."
   \[
   F_T = G + 2
   \]

Now, let's solve the equations:

From equation 3, we found \( G = 6 \).

Let's verify if this value is consistent with other equations:

- From equation 1:
  \[
  3G = 3C - 6 \implies 3(6) = 3C - 6 \implies 18 = 3C - 6 \implies 3C = 24 \implies C = 8
  \]

- From equation 2:
  \[
  3B + 2C = 59 \implies 3B + 2(8) = 59 \implies 3B + 16 = 59 \implies 3B = 43 \implies B = \frac{43}{3}
  \]
  (This value seems unusual, but let's proceed.)

- From equation 5:
  \[
  3B = 3F + 15 \implies \frac{43}{3} = F + 5 \implies F = \frac{43}{3} - 5 = \frac{43}{3} - \frac{15}{3}
  = \frac{28}{3}
  \]

- From equation 6:
  \[
  F_T = G + 2 \implies F_T = 6 + 2 = 8
  \]

The calculations are consistent with the given conditions. Therefore, the cost of a Greek salad at Texas BBQ
is:

Answer: 6
```

Mistake: At step 3, model calculated the value of "G", but from the start of the solution, "G" denotes the "cost of a Greek salad at Texas BBQ", not the cost of a Greek salad at Taste Good Cuisine.

C.2 o3-mini

Example 1: A pizza at Taste Good Cuisine and 3 lasagnas at Taste Good Cuisine cost 48 dollars. A lasagna at Urban Plate costs 6 dollars. 3 lasagnas at Urban Plate and 3 pizzas at Urban Plate cost 63 dollars. Question: how much does a lasagna at Taste Good Cuisine cost?

Model answer:

```
Let p be the price of a pizza at Taste Good Cuisine and l be the price of a lasagna at Taste Good Cuisine.
We are given:

1. p + 3l = 48.                    (Taste Good Cuisine)
```

```
2. A lasagna at Urban Plate costs 6 dollars.
3. 3 lasagnas at Urban Plate and 3 pizzas at Urban Plate cost 63 dollars.

From (3), the total cost of 3 lasagnas at Urban Plate is 3 × 6 = 18 dollars. This means that the 3 pizzas at
Urban Plate cost 63 - 18 = 45 dollars, so one pizza at Urban Plate costs 45/3 = 15 dollars.

A common (though unstated) assumption in these puzzles is that dishes with the same name cost the same
across restaurants. Thus, we assume that a pizza at Taste Good Cuisine also costs 15 dollars. Substitute
p = 15 into equation (1):

15 + 3l = 48
3l = 48 - 15 = 33
l = 33/3 = 11

Answer: 11
```

Mistake: The model made an assumption: "`A common (though unstated) assumption in these puzzles is that dishes with the same name cost the same across restaurants.`" However, this cannot be a mathematically valid assumption because following this assumption, it arrives at the conclusion that the price for lasagna at Urban Plate is not the same as that at Taste Good Cuisine (6 dollars versus 11 dollars).

We randomly selected 20 problems from the `ansDepth` $= 2$ set that triggered hallucinatory answer from o3-mini and manually checked the model answer, all of them shares this same mistake. Below is a similar example:

Example 2: A scrambled egg at Bistro Nice and a piece of cheese cake at Mike's Place cost 21 dollars. 2 scrambled eggs at Bistro Nice cost 6 dollars more than 2 pieces of cheese cake at Bistro Nice. A scrambled egg at Mike's Place costs 6 dollars. Question: how much does a scrambled egg at Bistro Nice cost?

Model answer:

```
Let S_B = price of a scrambled egg at Bistro Nice,
    C_B = price of a piece of cheesecake at Bistro Nice,
    C_M = price of a piece of cheesecake at Mike's Place.

The problem gives:
1. S_B + C_M = 21 dollars.
2. Two scrambled eggs at Bistro Nice cost 6 dollars more than two cheesecakes at Bistro Nice:
  2S_B = 2C_B + 6  →  S_B = C_B + 3.
3. A scrambled egg at Mike's Place costs 6 dollars (this is extra information).

Without further assumptions the system involves three unknowns (S_B, C_B, C_M) and only two equations.
However, it is common in these problems to assume that when the same item is mentioned (here "cheesecake")
it has a consistent price from one restaurant to the other. That is, we assume C_M = C_B.

Under that assumption the first equation becomes:
  S_B + C_B = 21.
Then using S_B = C_B + 3 we substitute:
  (C_B + 3) + C_B = 21  →  2C_B + 3 = 21  →  2C_B = 18  →  C_B = 9.
Thus, S_B = 9 + 3 = 12.

Answer: 12
```

Mistake: Same as above.

Example 3: 3 BLT sandwiches at Bistro Nice cost 6 dollars more than 3 pieces of cheese cake at Urban Plate. 3 burgers at Urban Plate cost 18 dollars less than 3 pieces of cheese cake at Bistro Nice. 2 BLT sandwiches at Urban Plate and 2 scrambled eggs at Urban Plate cost 34 dollars. A BLT sandwich at Urban Plate costs 8 dollars. 3 pieces of cheese cake at Bistro Nice cost 12 dollars more than 3 burgers at Bistro Nice. 2 burgers at Urban Plate cost 10 dollars less than 2 scrambled eggs at Bistro Nice. 3 pieces of cheese cake at Bistro Nice and a piece of cheese cake at Urban Plate cost 54 dollars. Question: how much does a piece of cheese cake at Urban Plate cost?

Model answer:

```
Let CC_U be the cost of one piece of cheese cake at Urban Plate.
```

```
Step 1. From “3 BLT sandwiches at Bistro Nice cost 6 dollars more than 3 pieces of cheese cake at Urban
Plate”:
  3·BLT_B = 3·CC_U + 6  →  BLT_B = CC_U + 2     (1)

Step 2. From “3 burgers at Urban Plate cost 18 dollars less than 3 pieces of cheese cake at Bistro Nice”:
  3·BUR_U = 3·CC_B - 18  →  With BUR_U determined later, we use this after finding BUR_U.

Step 3. “2 BLT sandwiches at Urban Plate and 2 scrambled eggs at Urban Plate cost 34 dollars”:
  2·BLT_U + 2·Eggs_U = 34  and we are given in Step 4 that BLT_U = 8.
  Thus, 2·8 + 2·Eggs_U = 34  →  16 + 2·Eggs_U = 34  →  2·Eggs_U = 18  →  Eggs_U = 9.

Step 4. “A BLT sandwich at Urban Plate costs 8 dollars”:
  BLT_U = 8.

Step 5. “3 pieces of cheese cake at Bistro Nice cost 12 dollars more than 3 burgers at Bistro Nice”:
  3·CC_B = 3·BUR_B + 12  →  CC_B = BUR_B + 4     (2)

Step 6. “2 burgers at Urban Plate cost 10 dollars less than 2 scrambled eggs at Urban Plate”:
  2·BUR_U = 2·Eggs_U - 10  →  2·BUR_U = 18 - 10  →  2·BUR_U = 8  →  BUR_U = 4.

Step 7. Now return to the burgers at Urban Plate in Step 2:
  3·BUR_U = 3·CC_B - 18  With BUR_U = 4,
  3·4 = 3·CC_B - 18  →  12 = 3·CC_B - 18  →  3·CC_B = 30  →  CC_B = 10.

Step 8. “3 pieces of cheese cake at Bistro Nice and a piece of cheese cake at Urban Plate cost 54 dollars”:
  3·CC_B + CC_U = 54  →  3·10 + CC_U = 54  →  30 + CC_U = 54  →  CC_U = 24.

The question asks: “how much does a piece of cheese cake at Urban Plate cost?”

Answer: 24
```

Mistake: `Step 6. “2 burgers at Urban Plate cost 10 dollars less than 2 scrambled eggs at Urban Plate”.` That is a hallucinatory condition. The actual condition is "2 burgers at Urban Plate cost 10 dollars less than 2 scrambled eggs at Bistro Nice".

WinSpot: GUI Grounding Benchmark with Multimodal Large Language Models

Zheng Hui, **Yinheng Li**, **Dan Zhao**, **Colby Banbury**, **Tianyi Chen**, **Kazuhito Koishida**

Columbia University, Microsoft

{zackhui, yinhengli, tianyi.chen, colbybanbury }@microsoft.com, {zh2483}@columbia.edu, {dz1158}@nyu.edu

Abstract

Graphical User Interface (GUI) automation relies on accurate GUI grounding. However, obtaining large-scale, high-quality labeled data remains a key challenge, particularly in desktop environments like Windows Operating System (OS). Existing datasets primarily focus on structured web-based elements, leaving a gap in real-world GUI interaction data for non-web applications. To address this, we introduce a new framework that leverages LLMs to generate large-scale GUI grounding data, enabling automated and scalable labeling across diverse interfaces. To ensure high accuracy and reliability, we manually validated and refined 5,000 GUI coordinate-instruction pairs, creating WinSpot—the first benchmark specifically designed for GUI grounding tasks in Windows environments. WinSpot provides a high-quality dataset for training and evaluating visual GUI agents, establishing a foundation for future research in GUI automation across diverse and unstructured desktop environments[1].

1 Introduction

Multimodal Large Language Models (MLLMs) exhibit impressive visual understanding and reasoning (Gandhi et al., 2023; Liu et al., 2024; Li et al., 2024; Zhang et al., 2025; Li et al., 2025b), enabling automation in complex real-world scenarios (Ai et al., 2024; Hui et al., 2025b). Among these, Graphical User Interface (GUI) automation has emerged as a critical application, where agents interpret on-screen elements and execute context-relevant actions for tasks such as software testing and application management (Yang et al., 2023a; Li et al., 2020; Wang et al., 2024b).

Despite significant advances in web and mobile GUI automation (Bavishi et al., 2023; Yang et al., 2023a; Cheng et al., 2024; Wang et al., 2024a; Hui et al., 2025a), GUI in Windows desktop environments remain largely unexplored, despite Windows system widespread use in professional and enterprise applications. This gap is particularly challenging because Windows applications lack a standardized UI representation such as HTML or DOM structures, requiring GUI grounding to rely purely on visual perception. Furthermore, Windows interfaces exhibit highly diverse layouts, where applications designed using different frameworks (e.g., Win32, UWP, Electron) follow inconsistent UI structures. Additionally, overlapping windows introduce ambiguity in detecting actionable elements, as interactable regions may be partially or fully obscured. Unlike web applications, where ARIA (Accessible Rich Internet Applications) attributes provide accessibility metadata, many Windows applications lack structured accessibility trees (a11y trees), making it difficult to extract UI component descriptions programmatically. Existing screenshot-based methods (Bavishi et al., 2023; Cheng et al., 2024) show promise but lack a large-scale, standardized benchmark specifically designed for Windows GUI automation. Without such a benchmark, researchers face challenges in systematically measuring progress, comparing approaches, and addressing the distinct complexities of desktop interfaces.

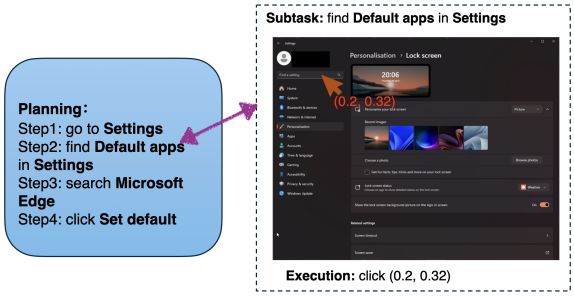


Figure 1: GUI grounding: locating actionable UI elements based on instructions.

To fill this void, we introduce **WinSpot**, a *large-*

[1]https://github.com/zackhuiiiii/WinSpot.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1086–1096
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

scale benchmark specifically designed for GUI grounding in Windows environments. Our main contributions are summarized below:

- **Two-Stage Labeling Framework.** We propose a scalable approach that utilizes MLLMs to generate coordinate-instruction pairs from diverse Windows screenshots, significantly reducing the initial labeling burden. Importantly, our method relies exclusively on raw screenshots, ensuring seamless adaptation across different Windows applications.
- **WinSpot: A First-of-its-Kind Windows GUI Benchmark.** Expanding on our two-stage framework, we introduce **WinSpot**—a comprehensive dataset with **over 5,000** human-validated coordinate-instruction pairs, covering diverse Windows environments, 21 times larger than previous benchmarks.

2 Related Work

2.1 UI Screen Understanding Dataset

A variety of datasets (Moran et al., 2018; He et al., 2021; Wu et al., 2023) have been developed to support UI modeling, primarily in the mobile domain. For instance, the AMP dataset (Zhang et al., 2021), containing 77k screens from 4,068 iOS apps. Another significant resource is Rico (Deka et al., 2017), the largest publicly available dataset for Android apps UI understanding. In the broader web and OS domain, datasets such as Mind2Web (Deng et al., 2024) , Visual-WebArena (Koh et al., 2024), and Windows Arena (Bonatti et al., 2024) offer simulated environments for various tasks. Existing GUI Grounding datasets overwhelmingly focus on mobile and web platforms, leaving desktop environments underexplored. The closest dataset related to desktop UI understanding is SeeClick (Cheng et al., 2024) and Os-atlas (Wu et al., 2024b), though it predominantly targets cross-platform settings and lacks a specific focus on Windows. Our work fills this gap by introducing a dataset tailored to desktop environments, particularly for Windows OS, which marks a novel contribution to the field.

3 Method

3.1 Data Construction

Unlike previous work (Cheng et al., 2024) that focuses on cross-domain tasks and structured data for training dataset construction, our approach targets the Windows OS (Figure 2). We propose

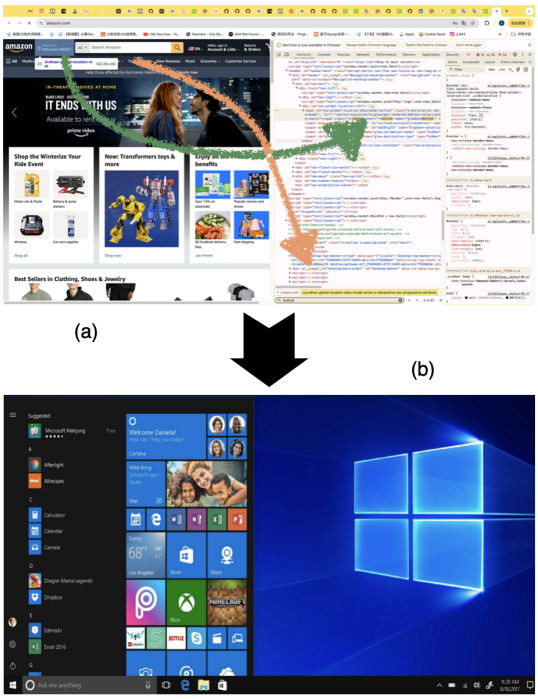


Figure 2: (a) Traditional methods rely on HTML or DOM files to locate icons during data construction. (b) Our proposed data alignment framework requires only raw screenshot images.

Instruction-Interactable Region Alignment (Figure 3), leveraging MLLMs to generate training data without HTML elements or accessibility trees. Since Windows applications lack a standardized UI representation and display diverse layouts across frameworks (e.g., Win32, UWP, Electron), our method relies entirely on visual cues for effective GUI grounding.

We first retrieve and filter images via the Bing API, then verify quality with Phi3-vision (Abdin et al., 2024b). Our goal is to collect diverse, representative Windows screenshots. We query screenshots of 700+ top apps from the Microsoft Store[2]. This model verifies resolution and screenshot validity. Quality-approved images are randomly added to our data bank. We expand our dataset via Bing API's similar image feature. Images failing quality checks are discarded.

Once filtered, we apply a proprietary Bert model with ViT encoder to perform icon grounding on the selected images. The ViT-Bert model generates bounding boxes around interactable icons in the images, which we use to create structured data. We then use GPT-4o for aligning the filtered images with corresponding icon descriptions. This align-

[2]https://www.microsoft.com/en-us/store/most-popular/apps/pc

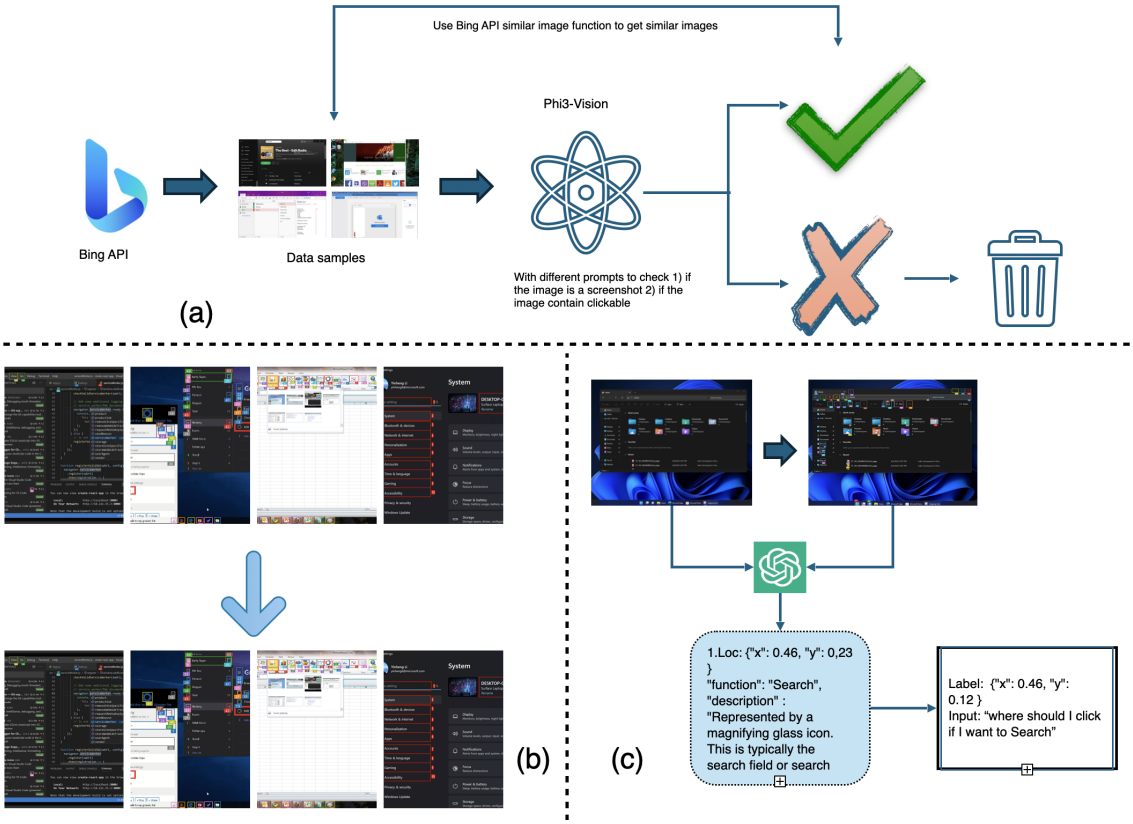


Figure 3: Overview of the Data Alignment Process: (a) illustrates the data filtering strategy using the Phi3 vision model, (b) shows the input and output of icon grounding with the in-house ViT-BERT model, and (c) the use of LLMs for GUI and description alignment.

ment process serves multiple purposes: 1) By using expensive models like GPT-4o only in the data alignment stage, we reduce computational costs while maintaining accuracy during the reasoning and inference processes. 2) Previous work (Zheng et al., 2024) has shown that providing GPT-4V with screenshots that only include bounding boxes and IDs can be misleading. The limited ability of GPT-4V to extract semantic information while predicting actions poses a challenge. To address this, our automated labeling pipeline incorporates semantic cues directly into the images during data construction, primarily by enhancing the prompts generated by GPT-4V. 3) By enriching the dataset with diverse semantic information, we ensure that the subsequent click agents can handle distributed tasks more effectively, improving overall performance. In addition to the data collected via the Bing API, we incorporated 500 images from CoVA dataset(Kumar et al., 2022), 500 images from WebSight dataset(Laurençon et al., 2024) to further enhance our training set. Result in a dataset around 60K to train our model. For more examples about the data construction, please refer to Appendix B.

3.2 WinSpot Benchmark

The *WinSpot* dataset consists of over 5,000 annotated [3] screenshots from 14 core Windows applications, each representing unique interaction types and layout structures. Examples from *WinSpot* are shown in Figure 4. The applications and their respective contributions to the dataset are shown in Figure 5. Each screenshot in *WinSpot* contains multiple interactable regions, such as buttons, menus, and icons, each annotated with its corresponding function. These annotations include bounding boxes around the interactable elements and their associated semantic descriptions, which are aligned with natural language instructions for both grounding and task prediction tasks. This variety ensures that *WinSpot* provides a challenging and comprehensive evaluation framework for GUI agents, enabling robust testing of both interaction precision and generalization across different applications.

WinSpot presents a diverse array of tasks, including file navigation, system settings adjustment, and text input, as well as more specialized tasks such

[3] More annotation detail in Appendix C

Figure 4: WinSpot examples

as process management in Task Manager and command execution in Command Prompt. These tasks encompass a wide range of complexity, from simple button clicks to more involved interactions that require an understanding of application-specific layouts. In addition to supporting GUI grounding tasks $P(y|S, x)$, *WinSpot* also be used in reverse tasks $P(x|S, y)$, where the model must predict the description of a given GUI element based on its location in the screenshot. This two-way task formulation enhances the evaluation by testing both the agent's understanding of visual cues and its ability to map interactions to the correct interface components.

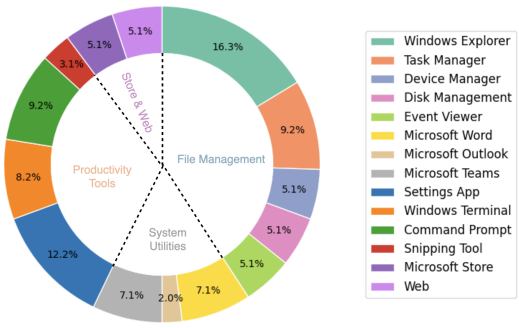


Figure 5: WinSpot Category

4 Experiments and Results

In this section, we evaluate both general-purpose models and GUI-specific models on our newly introduced **WinSpot** benchmark.

4.1 Baselines and Evaluation Metric

We compare multiple baselines, including general-purpose MLLMs (e.g., GPT-4o (OpenAI et al., 2024), GPT-4V (OpenAI et al., 2024), Phi3-Vision (Abdin et al., 2024a), MinGPT-v2 (Chen et al., 2023)) and GUI-focused models (e.g., Fuyu (Bavishi et al., 2023), CogAgent (Wang et al., 2024c), SeeClick (Cheng et al., 2024), Uground (Gou et al., 2024)). Consistent with prior studies on GUI grounding (Li et al., 2022; Yang et al., 2023b; Cheng et al., 2024), we adopt **click accuracy** as our primary metric. A prediction is considered correct if the model's predicted click coordinates fall within the bounding box of the ground-truth.

4.2 Results

Table 1 presents the click accuracy of various models across the four major subcategories of the WinSpot benchmark: File Management, System Settings, Productivity Tools, and MS Store & Web applications. These categories span a wide range of GUI interaction patterns allowing us to assess both generalization and domain sensitivity of each model. The best-performing model is **Uground**, which achieves a remarkable **44.2%** overall accuracy, significantly outperforming all other baselines. Its dominance is particularly evident in **System Settings** and **MS Store & Web**, where it reaches **51.4%** and **82.4%** respectively.

Among the general-purpose MLLMs, **GPT-4V** and **GPT-4o** show relatively higher click accuracy (**18.3%** and **16.5%**, respectively), with notable strengths in the **MS Store & Web** category—where visual layout conventions tend to be more standardized and semantically interpretable. This aligns with prior observations that LLMs pretrained on web data tend to generalize better in semi-structured interfaces but struggle with unstructured system UIs. However, their low scores in **System Settings** and **File Management** (e.g., GPT-4V: 6.3%, GPT-4o: 7.5%) reveal key limitations when navigating system-level layouts, likely due to the absence of such interfaces in their training data and the lack of spatial attention mechanisms specialized for desktop contexts. **Phi-3.5 Vision** and **MiniGPTv-2**, both smaller open-source models,

Method	Size	WinSpot				Total Task
		File Management	System	Productivity Tools	MS Store & Web	
MiniGPTv-2	7B	0%	0.6%	2.2%	5.8%	1.7%
GPT-4V	Unkown	8.0%	6.3%	15.6%	58.1%	18.3%
GPT-4o	Unkown	8.8%	7.5%	14.1%	47.7%	16.5%
Phi-3.5 Vision	4.2B	4.7%	5.8%	3.7%	25.5%	7.9%
Fuyu	8B	5.0%	9.2%	7.1%	34.3%	9.4%
CogAgent	18B	6.4%	8.1%	10.5%	60.8%	13.8%
SeeClick	9.6B	8.6%	16.6%	18.9%	70.6%	20.1%
GUIAct-Qwen	7B	10.8%	6.1%	13.6%	78.4%	18.0%
Uground	7B	27.2%	51.4%	45.4%	82.4%	44.2%

Table 1: Evaluation of Various Methods on WinSpot Subcategories

perform poorly across all subcategories, with overall accuracy below 10%. These results reinforce the importance of scale, training modality, and data coverage in grounding tasks. Interestingly, **Phi-3.5 Vision** performs slightly better in the **MS Store & Web** category, suggesting even smaller models can benefit from interface regularity if provided with sufficient multimodal alignment. Specialized GUI models such as **CogAgent**, **Fuyu**, **SeeClick**, and **GUIAct-Qwen** fall in an intermediate performance range (between **9.4%** and **20.1%** overall), with **SeeClick** standing out as the strongest among them. Notably, **SeeClick** attains **70.6%** accuracy on **MS Store & Web**, highlighting its suitability for commercial UI tasks, but only **16.6%** in **System Settings**, pointing to challenges in less standardized environments. Similarly, **GUIAct-Qwen** achieves competitive results in web-based domains but lacks consistency across system-heavy tasks, suggesting an over-reliance on pretraining priors that fail to capture the visual intricacies of Windows system utilities.

5 Discussion and Future Work

Our findings highlight a clear performance divide between general-purpose MLLMs and domain-specific GUI models. Generalist models such as GPT-4o and GPT-4V demonstrate only modest proficiency in GUI grounding, reflecting their limited pretraining exposure to Windows UI paradigms. Meanwhile, specialized models like Uground and SeeClick perform significantly better, particularly in structured tasks like web and app store interactions. However, even these tailored models struggle with system-level operations—such as task management, file navigation, or control panel interactions—where contextual reasoning and fine-grained spatial precision are required. This underscores a broader insight: current models, despite their size and multimodal capacity, lack robust spatial reasoning and memory mechanisms necessary for GUI automation in real-world settings. WinSpot helps uncover these limitations by evaluating not just interaction precision, but also the ability of models to align natural language instructions with semantically meaningful UI regions. Furthermore, this work is situated within a broader movement in NLP and multimodal learning: applying LLMs to real-world utility tasks beyond traditional text benchmarks. With growing industrial interest in automating workflows, testing software, and enabling human-in-the-loop systems, GUI agents will increasingly become critical enablers. Our benchmark and methodology lay the groundwork for these systems, while also exposing their current gaps. Going forward, we advocate for more research at the intersection of vision-language grounding, procedural planning (Li et al., 2025a), and user intent modeling. In particular, incorporating temporal dynamics (Jiang et al., 2025), multi-turn interactions (Liu et al., 2025), and UI state tracking may bridge the gap between static grounding and true GUI manipulation. Additionally, as LLM-driven agents are deployed in productivity tools, safety (Hui et al., 2024a; Zhang et al., 2024; Wu et al., 2024a) and interpretability will become pressing concerns—especially in high-stakes domains like healthcare, finance, and enterprise automation. In summary, WinSpot offers a much-needed benchmark for evaluating GUI grounding in Windows environments and serves as a testbed for the next generation of GUI agent. It pushes the research community to build models that are not only linguistically fluent but also visually and operationally grounded in environments where real users work.

6 Limitations

While WinSpot establishes a valuable benchmark for GUI grounding in Windows environments, several limitations remain. First, the dataset is derived mainly from a curated selection of popular Windows applications and supplemental sources, which may not capture the full diversity of desktop interfaces—especially those used in specialized or enterprise settings. Second, our automated labeling pipeline, although designed to reduce manual effort, relies on MLLMs for generating semantic cues; any inaccuracies or biases inherent in these models can propagate into the final annotations. Third, our evaluation metric, click accuracy, offers a simplified perspective on interaction performance, potentially overlooking nuanced aspects of user engagement such as multi-step workflows or gesture dynamics. Finally, the framework is optimized for static screenshots and may not generalize well to dynamic or adaptive interfaces that evolve in real time.

7 Ethical Considerations

In constructing WinSpot, we took deliberate steps to safeguard user privacy and ensure ethical data practices. All screenshots were rigorously reviewed and post-processed to remove personal or sensitive information before inclusion in the dataset. However, the selection process for source applications may introduce biases that affect the representativeness of the dataset. Additionally, the automated labeling pipeline's reliance on MLLMs could inadvertently propagate existing biases present in these models. We advocate for continuous audits and transparent documentation of both dataset composition and model performance, especially when these systems are applied in real-world scenarios such as automated testing or user assistance. As the deployment of GUI automation technology expands, it is imperative to consider the impact on user autonomy and employment, ensuring that such tools are used in a manner that respects consent, fairness, and accountability.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024a. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024b. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Lin Ai, Zheng Hui, Zizhou Liu, and Julia Hirschberg. 2024. Enhancing pre-trained generative language models with question attended span extraction on machine reading comprehension. In *In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 10046–10063, Miami, Florida, USA. Association for Computational Linguistics.*

Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. 2023. Fuyu-8b: A multimodal architecture for ai agents.

Rogerio Bonatti, Dan Zhao, Francesco Bonacci, Dillon Dupont, Sara Abdali, Yinheng Li, Yadong Lu, Justin Wagle, Kazuhito Koishida, Arthur Bucker, Lawrence Jang, and Zack Hui. 2024. Windows agent arena: Evaluating multi-modal os agents at scale. *Preprint*, arXiv:2409.08264.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoor-

thi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *Preprint*, arXiv:2310.09478.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. 2024. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9313–9332, Bangkok, Thailand. Association for Computational Linguistics.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pages 845–854.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2024. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36.

Kanishk Gandhi, Jan-Philipp Fraenken, Tobias Gerstenberg, and Noah Goodman. 2023. Understanding social reasoning in language models with language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 13518–13529. Curran Associates, Inc.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. 2024. Navigating the digital world as humans do: Universal visual grounding for gui agents. *arXiv preprint arXiv:2410.05243*.

Zecheng He, Srinivas Sunkara, Xiaoxue Zang, Ying Xu, Lijuan Liu, Nevan Wichers, Gabriel Schubiner, Ruby Lee, and Jindong Chen. 2021. Actionbert: Leveraging user actions for semantic understanding of user interfaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5931–5938.

Zheng Hui, Zhaoxiao Guo, Hang Zhao, Juanyong Duan, Lin Ai, Yinheng Li, Julia Hirschberg, and Congrui Huang. 2024a. Can open-source llms enhance data augmentation for toxic detection?: An experimental study. *arXiv preprint arXiv:2411.15175*.

Zheng Hui, Zhaoxiao Guo, Hang Zhao, Juanyong Duan, and Congrui Huang. 2024b. ToxiCraft: A novel framework for synthetic generation of harmful information. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16632–16647, Miami, Florida, USA. Association for Computational Linguistics.

Zheng Hui, Yinheng Li, Tianyi Chen, Colby Banbury, Kazuhito Koishida, et al. 2025a. Winclick: Gui grounding with multimodal large language models. *arXiv preprint arXiv:2503.04730*.

Zheng Hui, Xiaokai Wei, Yexi Jiang, Kevin Gao, Chen Wang, Frank Ong, Se eun Yoon, Rachit Pareek, and Michelle Gong. 2025b. Matcha: Can multi-agent collaboration build a trustworthy conversational recommender? *Preprint*, arXiv:2504.20094.

Yue Jiang, Jichu Li, Yang Liu, Dingkang Yang, Feng Zhou, and Quyu Kong. 2025. Danmakutppbench: A multi-modal benchmark for temporal point process modeling and understanding. *arXiv preprint arXiv:2505.18411*.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*.

Anurendra Kumar, Keval Morabia, William Wang, Kevin Chang, and Alex Schwing. 2022. CoVA: Context-aware visual attention for webpage information extraction. In *Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 80–90, Dublin, Ireland. Association for Computational Linguistics.

Hugo Laurençon, Léo Tronchon, and Victor Sanh. 2024. Unlocking the conversion of web screenshots into html code with the websight dataset. *Preprint*, arXiv:2403.09029.

Ao Li, Yuexiang Xie, Songze Li, Fugee Tsung, Bolin Ding, and Yaliang Li. 2025a. Agent-oriented planning in multi-agent systems. In *The Thirteenth International Conference on Learning Representations*.

Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. 2025b. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*.

Chengzu Li, Caiqi Zhang, Han Zhou, Nigel Collier, Anna Korhonen, and Ivan Vulić. 2024. TopViewRS: Vision-language models as top-view spatial reasoners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1786–1807, Miami, Florida, USA. Association for Computational Linguistics.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. 2022. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975.

Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. 2020. Mapping natural language instructions to mobile UI action sequences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8198–8210, Online. Association for Computational Linguistics.

Fuxiao Liu, Xiaoyang Wang, Wenlin Yao, Jianshu Chen, Kaiqiang Song, Sangwoo Cho, Yaser Yacoob, and Dong Yu. 2024. MMC: Advancing multimodal chart understanding with large-scale instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1287–1310, Mexico City, Mexico. Association for Computational Linguistics.

Yuhang Liu, Pengxiang Li, Zishu Wei, Congkai Xie, Xueyu Hu, Xinchen Xu, Shengyu Zhang, Xiaotian Han, Hongxia Yang, and Fei Wu. 2025. Infiguiagent: A multimodal generalist gui agent with native reasoning and reflection. *arXiv preprint arXiv:2501.04575*.

Kevin Moran, Carlos Bernal-Cárdenas, Michael Curcio, Richard Bonett, and Denys Poshyvanyk. 2018. Machine learning-based prototyping of graphical user interfaces for mobile apps. *IEEE Transactions on Software Engineering*, 46(2):196–221.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. 2024a. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. *arXiv preprint arXiv:2406.01014*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024b. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi

Li, Yuxiao Dong, Ming Ding, and Jie Tang. 2024c. Cogvlm: Visual expert for pretrained language models. *Preprint*, arXiv:2311.03079.

Chen Henry Wu, Jing Yu Koh, Ruslan Salakhutdinov, Daniel Fried, and Aditi Raghunathan. 2024a. Adversarial attacks on multimodal agents. *arXiv e-prints*, pages arXiv–2406.

Jason Wu, Siyan Wang, Siman Shen, Yi-Hao Peng, Jeffrey Nichols, and Jeffrey P Bigham. 2023. Webui: A dataset for enhancing visual ui understanding with web semantics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–14.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. 2024b. Osatlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023a. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *Preprint*, arXiv:2310.11441.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023b. The dawn of lmms: Preliminary explorations with gpt-4v(ision). *Preprint*, arXiv:2309.17421.

Xiaofeng Zhang, Fanshuo Zeng, Yihao Quan, Zheng Hui, and Jiawei Yao. 2025. Enhancing multimodal large language models complex reason via similarity computation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 10203–10211.

Xiaoyi Zhang, Lilian De Greef, Amanda Swearngin, Samuel White, Kyle Murray, Lisa Yu, Qi Shan, Jeffrey Nichols, Jason Wu, Chris Fleizach, et al. 2021. Screen recognition: Creating accessibility metadata for mobile applications from pixels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

Yanzhe Zhang, Tao Yu, and Diyi Yang. 2024. Attacking vision-language computer agents via pop-ups. *arXiv preprint arXiv:2411.02391*.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. Gpt-4v(ision) is a generalist web agent, if grounded. *Preprint*, arXiv:2401.01614.

A Training Details

To improve **WinClick**'s (Hui et al., 2025a) capacity for GUI understanding in Windows environments, we conducted extensive training using both **full fine-tuning** and **parameter-efficient tuning** via LoRA (Low-Rank Adaptation). This dual approach allowed us to explore trade-offs between performance and resource efficiency while maintaining compatibility with large-scale multimodal architectures.

For full fine-tuning, we updated the entire model, including the vision encoder, language decoder, and cross-modal attention layers. The visual encoder was initialized from a pre-trained ViT model and fine-tuned with an initial learning rate of 2×10^{-6}. The language backbone, based on Phi-3, was initialized with a learning rate of 5×10^{-6} and trained using a batch size of 32. A linear learning rate scheduler was applied with a warmup ratio of 0.03 to stabilize early training steps and mitigate gradient instability. The optimizer used was AdamW with weight decay set to 0.01.

For LoRA-based tuning, we injected low-rank matrices into the cross-modal attention layers and trained only these additional parameters, freezing the rest of the model. This method provided a significantly smaller memory footprint and faster training convergence while still yielding non-trivial performance improvements in grounding precision. LoRA ranks were set to 8 across all adapted modules, and dropout was applied with a probability of 0.1.

All experiments were conducted using 4 × NVIDIA H100 GPUs in a distributed setting using mixed-precision training (fp16). We used DeepSpeed and HuggingFace Accelerate to handle gradient accumulation, checkpointing, and parallelization. Training convergence was achieved within 5 epochs, with early stopping based on validation click accuracy on a held-out subset of WinSpot.

B More Training Data Construct Examples

To build a robust and diverse training corpus, our data pipeline aggregated screenshots from various sources, including real-world application states, software demos, and open benchmarks (e.g., CoVA (Kumar et al., 2022), WebSight (Laurençon et al., 2024)). After visual filtering and quality assessment using Phi3-Vision, selected images were passed through a multi-stage annotation pipeline. Figure 6 illustrates a sample of the training data. These examples span interaction types such as: Single-button confirmation dialogs (e.g., "Click OK to continue"), Multi-option menus (e.g., "Choose 'Save As' from the File menu"), Toolbar item selection (e.g., "Click the printer icon to print the document"), Search or input field interaction (e.g., "Type your query in the search box at the top right"). Each image contains between 5 interactable regions, and both the instruction and bounding box data were validated for semantic alignment by our human annotators (see Section C).

C Human Annotation

The annotation process follow similar settings as Hui et al. (2024b). For WinSpot involved a group of carefully selected annotators, all of whom were undergraduate, master's, or Ph.D. students, proficient in GUI operations and familiar with the Windows operating system. The annotation team consisted of individuals with diverse academic backgrounds, ensuring a broad understanding of GUI interactions across different applications. Each annotator was tasked with identifying and marking interactable regions within various Windows applications, focusing on elements such as buttons, icons, menus, and other clickable UI components. For the annotation process, annotators were provided with a set of Windows screenshots. These screenshots were then annotated using a custom tool that allowed them to create bounding boxes around each interactable element. Annotators were also required to provide corresponding descriptions of the elements, ensuring that both the visual and functional aspects of each UI component were documented. The entire annotation process was conducted in English to maintain consistency across all samples. To ensure data privacy, all screenshots were reviewed and post-processed to remove any personal information or sensitive content. The final dataset includes over 1,000 images and 5,000 instruction-click pairs, representing a comprehensive set of interactions across a variety of Windows applications.

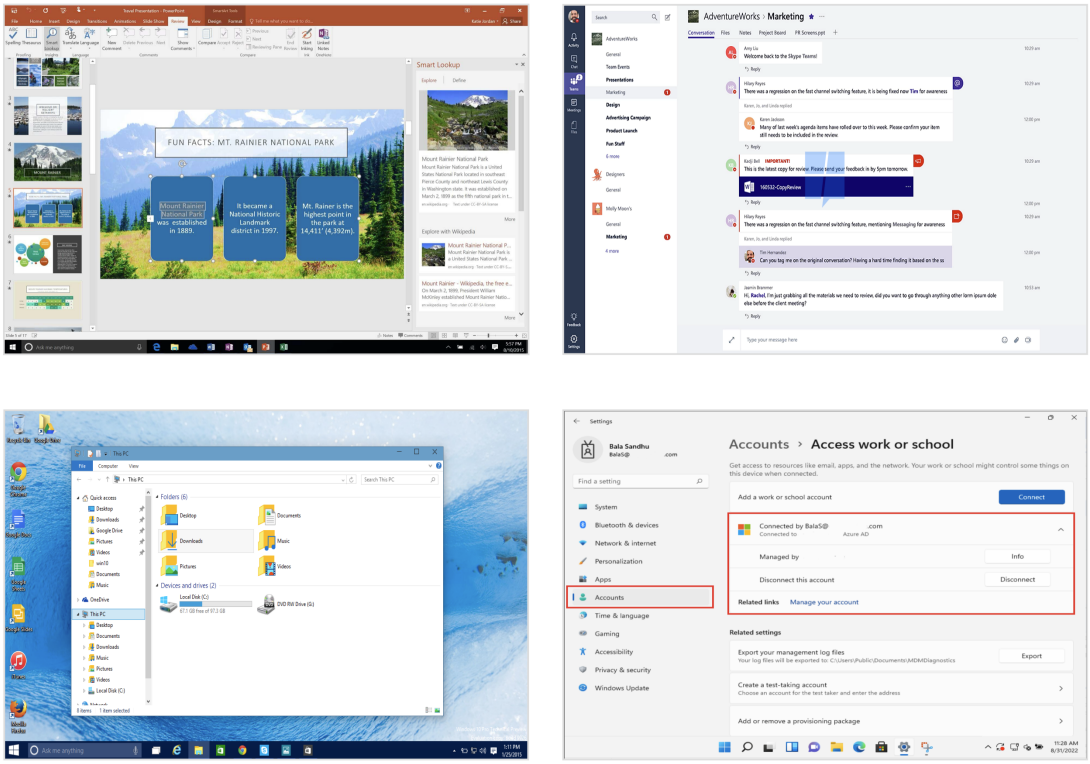


Figure 6: Examples of GUI grounding data generated during training set construction. Each box is annotated with the action-relevant region and its aligned instruction.

Spurious Correlations and Beyond: Understanding and Mitigating Shortcut Learning in SDOH Extraction with Large Language Models

Fardin Ahsan Sakib[1], **Ziwei Zhu**[1], **Karen Trister Grace**[2], **Meliha Yetişgen**[3], **Özlem Uzuner**[4]
[1]Department of Computer Science, [2]School of Nursing, [4]Department of Information Sciences and Technology
George Mason University
[3]Department of Biomedical Informatics & Medical Education, University of Washington
{fsakib,zzhu20,kgrace,ouzuner}@gmu.edu, melihay@uw.edu

Abstract

Social determinants of health (SDOH) extraction from clinical text is critical for downstream healthcare analytics. Although large language models (LLMs) have shown promise, they may rely on superficial cues leading to spurious predictions. Using the MIMIC portion of the SHAC (Social History Annotation Corpus) dataset and focusing on drug status extraction as a case study, we demonstrate that mentions of alcohol or smoking can falsely induce models to predict current/past drug use where none is present, while also uncovering concerning gender disparities in model performance. We further evaluate mitigation strategies—such as prompt engineering and chain-of-thought reasoning—to reduce these false positives, providing insights into enhancing LLM reliability in health domains.

1 Introduction

SDOH—including substance use, employment, and living conditions—strongly influence patient outcomes and clinical decision-making (Daniel et al., 2018; Himmelstein and Woolhandler, 2018; Armour et al., 2005). Extracting SDOH information from unstructured clinical text is increasingly important for enabling downstream healthcare applications and analysis (Jensen et al., 2012; Demner-Fushman et al., 2009). Although LLMs have shown promise in clinical natural language processing (NLP) tasks (Hu et al., 2024; Liu et al., 2023; Singhal et al., 2023), they often rely on superficial cues (Tang et al., 2023; Zhao et al., 2017), potentially leading to incorrect predictions undermining trust and utility in clinical settings.

Recent work has highlighted how LLMs can exhibit "shortcut learning" behaviors (Tu et al., 2020; Ribeiro et al., 2020; Zhao et al., 2018), where they exploit spurious patterns in training data rather than learning causal, generalizable features. This phenomenon spans various NLP tasks, from natural language inference (McCoy et al., 2019) to question-answering (Jia and Liang, 2017), and in clinical domains can lead to incorrect assumptions about patient conditions (Brown et al., 2023; Jabbour et al., 2020), threatening the utility of automated systems.

We investigate how LLMs produce spurious correlations in SDOH extraction through using drug status time classification (current, past, or none/unknown) as a case study. Using the MIMIC (Johnson et al., 2016) portion of the SHAC (Lybarger et al., 2021) dataset, we examine zero-shot and in-context learning scenarios across multiple LLMs (Llama (AI, 2024), Qwen (Yang et al., 2024), Llama3-Med42-70B (Christophe et al., 2024)).We explore multiple mitigation strategies to address these spurious correlations: examining the causal role of triggers through controlled removal experiments, implementing targeted prompt engineering approaches like chain-of-thought (CoT) reasoning (Wei et al., 2022), incorporating warning-based prompts, and augmenting with additional examples. While these interventions show promise—significant false positive rates persist, highlighting the deep-rooted nature of these biases and the need for more sophisticated solutions.

Contributions:

1. We present the first comprehensive analysis of spurious correlations in SDOH extraction across multiple LLM architectures, including domain-specialized models. Through extensive experiments in zero-shot and ICL settings, we demonstrate how models rely on superficial cues and verify their causal influence through controlled ablation studies.

2. We uncover systematic gender disparities in model performance, demonstrating another form of spurious correlation where models inappropriately leverage patient gender for drug

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1097–1106
July 27 - August 1, 2025

status time classification predictions.

3. We evaluate multiple prompt-based mitigation strategies (CoT, warnings, more examples) and analyze their limitations, demonstrating that while they reduce incorrect drug status time predictions, more robust solutions are needed for reliable clinical NLP deployments.

2 Related Work

Previous work on extracting SDOH from clinical text spans a progression from rule-based methods to fine-tuned neural models, leveraging annotated corpora for tasks like substance use and employment status extraction (Hatef et al., 2019; Patra et al., 2021; Yu et al., 2022; Han et al., 2022; Uzuner et al., 2008; Stemerman et al., 2021; Lybarger et al., 2023). More recent efforts have explored prompt-based approaches with LLMs, including GPT-4, to reduce reliance on extensive annotations (Ramachandran et al., 2023). While these approaches achieve competitive performance, studies across NLP tasks have shown that both fine-tuned and prompting-based methods often exploit spurious correlations or superficial cues (Ribeiro et al., 2020; Geirhos et al., 2020; Tu et al., 2020). Prior investigations have focused largely on spurious correlations in standard NLP tasks and supervised scenarios (McCoy et al., 2019; Zhao et al., 2018). In contrast, our work examines how these issues manifest in zero-shot and in-context SDOH extraction settings, and we propose prompt-level strategies to mitigate these correlations.

3 Methodology

3.1 Dataset and Task

We use the MIMIC-III portion of the SHAC dataset (Lybarger et al., 2021), which comprises 4405 deidentified social history note sections derived from MIMIC-III (Johnson et al., 2016) and the University of Washington clinical notes. SHAC is annotated using the BRAT tool (Stenetorp et al., 2012), capturing a variety of SDOH event types (e.g., Alcohol, Drug, Tobacco) as triggers along with associated arguments, including temporal status. To enable demographic analysis, we augmented the SHAC data by linking it with patient demographic information available in the original MIMIC-III dataset.

In this work, we examine spurious correlations in SDOH extraction through temporal drug status classification (current, past, or none/unknown). We adopt a two-step pipeline (Ma et al., 2022, 2023):

(1) Trigger Identification: Given a social history note, the model identifies spans corresponding to the target event type (e.g.,drug use).

(2) Argument Resolution: For each trigger, models applies a multiple-choice QA prompt to determine the temporal status (current/past/none).

The dataset contains diverse patterns of substance documentation, see Appendix B for detailed examples of the task and annotation schema.

3.2 Experimental Setup

Model Configurations We evaluate multiple model configurations:

- **Zero-Shot:** Models receive only task instructions and input text, with no examples
- **In-Context Learning (ICL):** Models are provided with three example demonstrations before making predictions on a new instance. Examples are selected to maintain balanced representation across substance use patterns (none/single/multiple) and drug use outcomes (positive/negative).
- **Fine-Tuning (SFT):** We also fine-tune a Llama-3.1-8B model on the MIMIC portion of the SHAC dataset to assess whether domain adaptation reduces spurious correlations.

We consider Llama-3.1-70B (zero-shot, ICL), Llama-3.1-8B (fine-tuned on MIMIC), Qwen-72B (ICL), and Llama3-Med42-70B (ICL). These models span various parameter sizes and domain specializations. The fine-tuned Llama-8B model provides insights into whether in-domain adaptation mitigates the observed shortcut learning.

Prompting Strategies We additionally use two other prompting strategies (see Appendix A for complete templates):

Chain-of-Thought (CoT): This prompt explicitly guides reasoning through five steps: (1) read the social history note carefully, (2) identify relevant information, (3) consider examples provided, (4) explain your reasoning process, (5) provide the answer. This encourages explicit reasoning to reduce shortcuts.

Warning-Based: This incorporates explicit guidelines to counter spurious correlations: (1) evaluate each factor independently - never assume

one behavior implies another, (2) extract only explicitly stated information - avoid making assumptions based on demographic or other factors, (3) use [none] when information isn't mentioned.

Evaluation Framework Our primary evaluation metric is the false positive rate (FPR), defined as: $FPR = FP/(FP + TN)$ where FP represents false positives (predicted current/past use when ground truth was none/unknown) and TN represents true negatives (correctly predicted none/unknown). We prioritize FPR given the clinical risks of incorrect positive drug use predictions—including patient stigmatization, biased provider perceptions, and diminished trust in automated systems (Van Boekel et al., 2013; Dahl et al., 2022). A higher FPR indicates more frequent erroneous predictions that could directly impact patient care. We specifically examine FPR disparities between substance-positive and substance-negative contexts to reveal whether models rely on superficial cues rather than actual evidence. See Appendix C for extended discussion.

To analyze potential spurious correlations, we categorize notes based on their ground truth substance use status:

- **Substance-positive**: Notes documenting current/past use of the respective substance (alcohol or smoking)
- **Substance-negative**: Notes where the ground truth indicates no use or unknown status

Experimental Settings

- **Original:** Evaluate models on the original notes.
- **Without Alcohol/Smoking Triggers:** Remove mentions of alcohol/smoking to test their causal role in inducing false positives.

4 Results

4.1 RQ1: Do Large Language Models Exhibit Spurious Correlations in SDOH Extraction?

As shown in Table 1, our analysis in a zero-shot setting with Llama-70B reveals high false positive rates for drug status time classification in alcohol-positive (66.21%) and smoking-positive (61.11%) notes. In contrast, alcohol-negative and smoking-negative notes show substantially lower false positive rates (28.83% and 29.76%, respectively). This stark contrast suggests that the mere presence of alcohol or smoking triggers biases the model towards inferring nonexistent drug use. These biases likely stem from the pre-training phase, potentially reinforcing societal assumptions about correlations between different types of substance use.

4.2 RQ2: Do In-Context Learning and Fine-Tuning Reduce These Spurious Correlations?

Providing three in-context examples reduces false positives significantly. For Llama-70B, ICL lowers alcohol-positive mismatches from 66.21% to 48.28%, though a gap remains relative to alcohol-negative notes (11.71%). Similarly, smoking-positive mismatches decrease from 61.11% to 36.42% versus 18.05% for smoking-negative. The effectiveness of ICL suggests that explicit examples help the model focus on relevant features, though the persistence of some bias indicates deep-rooted associations from pre-training. Fine-tuning Llama-8B on the MIMIC subset (SFT) yields further improvements: alcohol-positive mismatches drop to 32.41% and smoking-positive to 36.42%, with corresponding negatives at 12% and 7% respectively, indicating that domain adaptation helps override some pre-trained biases.

4.3 RQ3: Are These Superficial Mentions Causally Driving the Model's Predictions?

To confirm the causal role of alcohol and smoking mentions, we remove these triggers from the notes. Across models, this consistently lowers false positives. For instance, Llama-70B zero-shot sees alcohol-positive mismatches fall from 66.21% to 55.17% after removing alcohol triggers. Similarly, Llama-8B-SFT reduces alcohol-positive errors from 32.41% to 26.9%. Similar trends are observed across other architectures including domain-specific models (see appendix E), confirming that alcohol and smoking cues spuriously bias the models' drug-use predictions.

4.4 RQ4: Are there systematic demographic variations in these spurious correlations?

Beyond substance-related triggers, our analysis (Table 2) uncovers another concerning form of spurious correlation: systematic performance differences based on patient gender. Just as models incorrectly rely on mere mentions of alcohol or smoking to infer substance use, they appear to

Table 1: False Positive Rates (%) Across Different Models and Approaches. *Smoking+Alcohol* refers to cases where both *Smoking-positive* and *Alcohol-positive* are true.

Cases	Llama-70B					Llama-8B		Llama3-Med42-70B	Qwen-72B
	Zero-shot	ICL	CoT	Warning	Increased-Examples	Vanilla	Fine-tuned	ICL	ICL
Alcohol-positive	66.21	48.28	33.79	40.69	45.52	73.10	32.41	66.90	62.76
Smoking-positive	61.11	36.42	25.93	29.63	30.25	74.07	36.42	57.41	53.09
Alcohol-negative	28.83	11.71	6.76	5.41	10.81	37.39	12.16	16.22	46.85
Smoking-negative	29.76	18.05	10.73	11.22	20.00	33.66	7.32	19.51	53.17
Smoking+Alcohol	73.26	51.16	34.88	45.35	39.53	81.40	40.70	76.74	56.98

leverage patient gender as an inappropriate predictive signal. For the base Llama-70B model in zero-shot settings, false positive rates show stark gender disparities - male patients consistently face higher misclassification rates compared to female patients (71.15% vs 53.66% for alcohol-positive cases, and 66.67% vs 50.88% for smoking-positive cases). This pattern persists with in-context learning, with the gender gap remaining substantial (alcohol-positive: 52.88% male vs 36.59% female). Fine-tuned models showed similar disparities, with Llama-8B-SFT maintaining a performance gap of approximately 15 percentage points between genders for alcohol-positive cases.

Notably, these gender-based differences exhibit complex interactions with substance-related triggers. Cases involving positive substances mentions show the most pronounced disparities, with male patients seeing up to 20 percentage point higher false positive rates. This suggests that the model's shortcut learning compounds across different dimensions - gender biases amplify substance-related biases and vice versa. The persistence of these interacting biases across model architectures, sizes, and prompting strategies suggests they arise from deeply embedded patterns in both pre-training data and medical documentation practices.

5 Mitigation Strategies and Results

We explore several mitigation techniques to address the spurious correlations identified in our analysis:

Chain-of-Thought (CoT) As shown in Table 1, instructing the model to reason step-by-step before producing an answer leads to substantial reductions across all architectures. For Llama-70B, CoT reduces alcohol-positive mismatches from 66.21% (zero-shot) to 33.79%, with smoking-positive cases decreasing from 61.11% to 25.93%. Similar improvements are observed in other models (see appendix F), with Qwen-72B showing particularly strong response to CoT. This suggests CoT helps models avoid superficial cues and focus on explicit information.

Warning-Based Instructions We prepend explicit instructions cautioning the model not to assume drug use without evidence and to treat each factor independently. With Llama-70B, these warnings lower alcohol-positive mismatches from 66.21% to approximately 40.69%, and also benefit smoking-positive scenarios. While not as strong as CoT, these warnings yield meaningful improvements across different architectures.

Increased Number of Examples Providing more than three examples—up to eight—further stabilizes predictions. For Llama-70B, increasing the number of examples reduces false positive rates considerably, with alcohol-positive mismatches falling to 45.52% (compared to 66.21% zero-shot). Similar trends are observed in other models, though the magnitude of improvement varies (see appendix F). While not as dramatic as CoT, additional examples help guide models away from faulty heuristics.

6 Discussion

Our findings highlight a key challenge in applying large language models to clinical information extraction: even when models achieve strong performance on average, they rely on superficial cues rather than genuine understanding of the underlying concepts. The presence of alcohol- or smoking-related mentions biases models to infer drug use incorrectly, and these shortcuts persist across Llama variants, Qwen, and Llama3-Med42-70B. The effectiveness of mitigation strategies like chain-of-thought reasoning, warning-based instructions, and additional examples underscores the importance of careful prompt design. While these interventions help guide models to focus on explicit evidence, their partial success suggests the need for more

Table 2: Gender-Based Analysis of False Positive Rates (%) Across Models

	Llama-70B Zero-shot		Llama-70B ICL		Llama-8B SFT		Qwen-72B	
Cases	Female	Male	Female	Male	Female	Male	Female	Male
Alcohol-positive	53.66	71.15	36.59	52.88	21.95	36.54	68.29	60.58
Smoking-positive	50.88	66.67	28.07	40.95	24.56	42.86	49.12	55.24
Alcohol-negative	29.13	28.42	9.45	14.74	9.45	15.79	47.24	46.32
Smoking-negative	27.03	32.98	9.91	27.66	6.31	8.51	54.05	52.13
Smoking+Alcohol	81.82	84.62	54.55	58.97	27.27	53.85	27.27	30.77

robust approaches - integrating domain-specific knowledge, implementing adversarial training, or curating more balanced datasets. Our demographic analysis reveals that these spurious correlations are not uniformly distributed across patient groups, raising fairness concerns for clinical deployment. Addressing such disparities requires both algorithmic improvements and careful consideration of deployment strategies. Clinicians and stakeholders must be aware of these limitations before deploying LLMs in clinical decision-support systems. Understanding these systematic biases in automated analysis can inform improvements not only in model development but also in clinical documentation practices and standards.

7 Implications Beyond NLP: Clinical Documentation and Practice

The implications of this study extend beyond NLP methodologies. Our analysis reveals that these models not only learn but potentially amplify existing biases in clinical practice. The identified error patterns—particularly the tendency to infer substance use from smoking/alcohol mentions and gender-based performance disparities—mirror documented provider biases in clinical settings (Saloner et al., 2023; Meyers et al., 2021). Notably, these biases appear to originate partly from medical documentation practices themselves (Ivy et al., 2024; Kim et al., 2021; Markowitz, 2022). Our finding that explicit evidence-based reasoning (through CoT) reduces these biases aligns with established strategies for mitigating provider bias (Mateo and Williams). This parallel between computational and human biases suggests that systematic analysis of LLM behavior could inform broader efforts to identify and address biases in medical documentation and practice, potentially contributing to improved provider education and documentation standards.

8 Conclusion

This work presents the first systematic exploration of spurious correlations in SDOH extraction, revealing how contextual cues can lead to incorrect and potentially harmful predictions in clinical settings. Beyond demonstrating the problem, we've evaluated several mitigation approaches that, while promising, indicate the need for more sophisticated solutions. Future work should focus on developing robust debiasing techniques, leveraging domain expertise, and establishing comprehensive evaluation frameworks to ensure reliable deployment across diverse populations.

9 Limitations

Dataset limitations Our analysis relied exclusively on the MIMIC portion of the SHAC dataset, which constrains the generalizability of our findings. While we observe consistent gender-based performance disparities, a more diverse dataset could help establish the breadth of these biases.

Model coverage We focused solely on open-source large language models (e.g., Llama, Qwen). Extending the evaluation to additional data sources, closed-source models, and other domain-specific architectures would help verify the robustness of our conclusions.

Causal understanding While we established the causality of triggers through removal experiments, understanding why specific triggers affect certain models or scenarios would require deeper analysis using model interpretability techniques.

Methodology scope Our study focused exclusively on generative methods; results may not generalize to traditional pipeline-based approaches that combine sequence labeling and relation classification.

Mitigation effectiveness While we identified various spurious correlations, our mitigation strategies

could not completely address the problem, leaving room for future work on addressing these issues.

10 Ethics Statement

All experiments used de-identified social history data from the SHAC corpus, with LLMs deployed on a secure university server. We followed all data use agreements and institutional IRB protocols. Although the dataset is fully de-identified, biases within the models could raise ethical concerns in real-world applications. Further validation and safeguards are recommended before clinical deployment.

11 Acknowledgments

We thank our collaborators for their valuable feedback and support. Generative AI assistants were used for grammar checking and LaTeX formatting; the authors retain full responsibility for the final content and analysis.

References

Meta AI. 2024. Llama 3.1 model card. https://github.com/meta-llama/llama-models/blob/main/models/llama3_1/MODEL_CARD.md. Accessed: 2024-12-13.

BS Armour, T Woollery, A Malarcher, TF Pechacek, and C Husten. 2005. Annual smoking-attributable mortality, years of potential life lost, and productivity losses—united states, 1997-2001. *JAMA: Journal of the American Medical Association*, 294(7).

Alexander Brown, Nenad Tomasev, Jan Freyberg, Yuan Liu, Alan Karthikesalingam, and Jessica Schrouff. 2023. Detecting shortcut learning for fair medical ai using shortcut testing. *Nature communications*, 14(1):4314.

Clément Christophe, Praveen K Kanithi, Tathagata Raha, Shadab Khan, and Marco AF Pimentel. 2024. Med42-v2: A suite of clinical llms. *Preprint*, arXiv:2408.06142.

Rachel A Dahl, J Priyanka Vakkalanka, Karisa K Harland, and Joshua Radke. 2022. Investigating healthcare provider bias toward patients who use drugs using a survey-based implicit association test: Pilot study. *Journal of addiction medicine*, 16(5):557–562.

Hilary Daniel, Sue S Bornstein, Gregory C Kane, Health, and Public Policy Committee of the American College of Physicians*. 2018. Addressing social determinants to improve patient care and promote health equity: an american college of physicians position paper. *Annals of internal medicine*, 168(8):577–578.

Dina Demner-Fushman, Wendy W Chapman, and Clement J McDonald. 2009. What can natural language processing do for clinical decision support? *Journal of biomedical informatics*, 42(5):760–772.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Sifei Han, Robert F Zhang, Lingyun Shi, Russell Richie, Haixia Liu, Andrew Tseng, Wei Quan, Neal Ryan, David Brent, and Fuchiang R Tsui. 2022. Classifying social determinants of health from unstructured electronic health records using deep learning-based natural language processing. *Journal of biomedical informatics*, 127:103984.

Elham Hatef, Masoud Rouhizadeh, Iddrisu Tia, Elyse Lasser, Felicia Hill-Briggs, Jill Marsteller, Hadi Kharrazi, et al. 2019. Assessing the availability of data on social and behavioral determinants in structured and unstructured electronic health records: a retrospective analysis of a multilevel health care system. *JMIR medical informatics*, 7(3):e13802.

David U Himmelstein and Steffie Woolhandler. 2018. Determined action needed on social determinants. *Annals of internal medicine*, 168(8):596–597.

Yan Hu, Qingyu Chen, Jingcheng Du, Xueqing Peng, Vipina Kuttichi Keloth, Xu Zuo, Yujia Zhou, Zehan Li, Xiaoqian Jiang, Zhiyong Lu, et al. 2024. Improving large language models for clinical named entity recognition via prompt engineering. *Journal of the American Medical Informatics Association*, page ocad259.

Zalaya K Ivy, Sharon Hwee, Brittany C Kimball, Michael D Evans, Nicholas Marka, Catherine Bendel, and Alexander A Boucher. 2024. Disparities in documentation: evidence of race-based biases in the electronic medical record. *Journal of Racial and Ethnic Health Disparities*, pages 1–7.

Sarah Jabbour, David Fouhey, Ella Kazerooni, Michael W Sjoding, and Jenna Wiens. 2020. Deep learning applied to chest x-rays: exploiting and preventing shortcuts. In *Machine Learning for Healthcare Conference*, pages 750–782. PMLR.

Peter B Jensen, Lars J Jensen, and Søren Brunak. 2012. Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*, 13(6):395–405.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Min Kyung Kim, Joy Noel Baumgartner, Jennifer Headley, Julius Kirya, James Kaggwa, and Joseph R Egger. 2021. Medical record bias in documentation of obstetric and neonatal clinical quality of care indicators in uganda. *Journal of Clinical Epidemiology*, 136:10–19.

Zhengliang Liu, Yue Huang, Xiaowei Yu, Lu Zhang, Zihao Wu, Chao Cao, Haixing Dai, Lin Zhao, Yiwei Li, Peng Shu, et al. 2023. Deid-gpt: Zero-shot medical text de-identification by gpt-4. *arXiv preprint arXiv:2303.11032*.

Kevin Lybarger, Nicholas J Dobbins, Ritche Long, Angad Singh, Patrick Wedgeworth, Özlem Uzuner, and Meliha Yetisgen. 2023. Leveraging natural language processing to augment structured social determinants of health data in the electronic health record. *Journal of the American Medical Informatics Association*, 30(8):1389–1397.

Kevin Lybarger, Mari Ostendorf, and Meliha Yetisgen. 2021. Annotating social determinants of health using active learning, and characterizing determinants using neural event extraction. *Journal of Biomedical Informatics*, 113:103631.

Mingyu Derek Ma, Alexander K Taylor, Wei Wang, and Nanyun Peng. 2022. Dice: data-efficient clinical event extraction with generative models. *arXiv preprint arXiv:2208.07989*.

Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! *arXiv preprint arXiv:2303.08559*.

David M Markowitz. 2022. Gender and ethnicity bias in medicine: A text analysis of 1.8 million critical care records. *PNAS nexus*, 1(4):pgac157.

CM Mateo and DR Williams. Addressing bias and reducing discrimination. *The professional responsibility of health care providers*, 2020:95.

RT McCoy, E Pavlick, and T Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. arxiv preprint arxiv: 190201007.

SA Meyers, VA Earnshaw, Brittany D'Ambrosio, Natasia Courchesne, Dan Werb, and LR Smith. 2021. The intersection of gender and drug use-related stigma: A mixed methods systematic review and synthesis of the literature. *Drug and alcohol dependence*, 223:108706.

Braja G Patra, Mohit M Sharma, Veer Vekaria, Prakash Adekkanattu, Olga V Patterson, Benjamin Glicksberg, Lauren A Lepow, Euijung Ryu, Joanna M Biernacka, Al'ona Furmanchuk, et al. 2021. Extracting social determinants of health from electronic health records using natural language processing: a systematic review. *Journal of the American Medical Informatics Association*, 28(12):2716–2727.

Giridhar Kaushik Ramachandran, Yujuan Fu, Bin Han, Kevin Lybarger, Nicholas J Dobbins, Özlem Uzuner, and Meliha Yetisgen. 2023. Prompt-based extraction of social determinants of health using few-shot learning. *Preprint*, arXiv:2306.07170.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*.

Brendan Saloner, Wenshu Li, Michael Flores, Ana M Progovac, and Benjamin Lê Cook. 2023. A widening divide: Cigarette smoking trends among people with substance use disorder and criminal legal involvement: Study examines cigarette smoking trends among people with substance use disorders and people with criminal legal involvement. *Health Affairs*, 42(2):187–196.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Rachel Stemerman, Jaime Arguello, Jane Brice, Ashok Krishnamurthy, Mary Houston, and Rebecca Kitzmiller. 2021. Identification of social determinants of health using multi-label classification of electronic health record clinical notes. *JAMIA open*, 4(3):ooaa069.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.

Ruixiang Tang, Dehan Kong, Longtao Huang, and Hui Xue. 2023. Large language models can be lazy learners: Analyze shortcuts in in-context learning. *arXiv preprint arXiv:2305.17256*.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Özlem Uzuner, Ira Goldstein, Yuan Luo, and Isaac Kohane. 2008. Identifying patient smoking status from medical discharge records. *Journal of the American Medical Informatics Association*, 15(1):14–24.

Leonieke C Van Boekel, Evelien PM Brouwers, Jaap Van Weeghel, and Henk FL Garretsen. 2013. Stigma among health professionals towards patients with substance use disorders and its consequences for healthcare delivery: systematic review. *Drug and alcohol dependence*, 131(1-2):23–35.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou,

et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Zehao Yu, Xi Yang, Chong Dang, Songzi Wu, Prakash Adekkanattu, Jyotishman Pathak, Thomas J George, William R Hogan, Yi Guo, Jiang Bian, et al. 2022. A study of social and behavioral determinants of health in lung cancer patients using transformers-based natural language processing models. In *AMIA Annual Symposium Proceedings*, volume 2021, page 1225.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876*.

A Prompting Strategies

All prompting approaches share a base system message identifying the model's role as "an AI assistant specialized in extracting and analyzing social history information from medical notes." Each strategy then builds upon this foundation with specific modifications:

Zero-Shot

The baseline approach uses a minimal prompt structure: System: AI assistant specialized in social history extraction User: For the following social history note: [Clinical note text] [Task instruction] [Options if applicable] This setup evaluates the model's ability to perform extraction tasks using only its pre-trained knowledge, without additional guidance or examples.

In-Context Learning (ICL)

This approach augments the base prompt with three carefully selected demonstration examples. Each example follows a structured JSON format: json "id": "example-id", "instruction": "Extract all Drug text spans...", "input": "Social History: Patient denies drug use...", "options": "[Multiple choice options if applicable]", "output": "Expected extraction or classification"

Chain-of-Thought (CoT)

Building upon ICL, this method explicitly guides the model through a structured reasoning process: Please approach this task step-by-step: 1. Carefully read the social history note 2. Identify all relevant information related to the question 3. Consider the examples provided 4. Explain your reasoning process 5. Provide your final answer This approach aims to reduce spurious correlations and shortcut learning by encouraging explicit articulation of the reasoning process before arriving at the final extraction or classification.

Warning-Based

This specialized approach incorporates explicit rules and warnings in the system message: Important Guidelines: 1. Evaluate each factor independently - never assume one behavior implies another 2. Extract only explicitly stated information - don't make assumptions based on demographics or other factors 3. If information isn't mentioned, use [none] or select "not mentioned" option These guidelines specifically address the challenge of false positives in substance use detection by discouraging inference-based conclusions without explicit textual evidence. The warnings are designed to counteract the model's tendency to make assumptions based on superficial cues or demographic factors.

B Dataset Details

B.1 Data Format and Annotation Process

The SHAC dataset originally consists of paired text files (.txt) containing social history notes and annotation files (.ann) capturing SDOH information. We convert these into a question-answering format to evaluate LLMs. Below we demonstrate this process with a synthetic example:

Raw Note (.txt)

```
SOCIAL HISTORY:
```

```
Patient occasionally uses alcohol.
Denies any illicit drug use.
```

BRAT Annotations (.ann)

```
T1 Alcohol 24 31 alcohol
T2 Drug 47 50 drug
T3 StatusTime 8 19 occasionally
T4 StatusTime 32 37 denies

E1 Alcohol:T1 Status:T3
E2 Drug:T2 Status:T4

A1 StatusTimeVal T3 current
A2 StatusTimeVal T4 none
```

Here, T1 and T2 are triggers - spans of text that indicate the presence of SDOH events (e.g., "alcohol" for substance use). The annotations also capture arguments - additional information about these events, such as their temporal status represented by T3 and T4. For example, T3 ("occasionally") indicates a temporal status of *current* for alcohol use.

We transform these structured annotations into two types of questions:

Trigger Identification Questions about identifying relevant event spans:

```
{"id": "0001-Alcohol",
 "instruction": "Extract all Alcohol
  text spans as it is from the note.
  If multiple spans present, separate
  them by [SEP]. If none, output
  [none].",
 "input": "SOCIAL HISTORY: Patient
  occasionally uses alcohol. Denies
  any illicit drug use.",
 "output": "alcohol"}
```

Argument-Resolution Questions about determining event properties:

```
{"id": "0001-Alcohol_StatusTime",
 "instruction": "Choose the best
  StatusTime value for the <alcohol>
  (Alcohol) from the note:",
 "input": "SOCIAL HISTORY: Patient
  occasionally uses alcohol. Denies
  any illicit drug use.",
 "options": "Options: (a) none.
  (b) current. (c) past.
  (d) Not Applicable.",
 "output": "(b) current."}
```

C Metric Selection and Justification

Our focus on False Positive Rate (FPR) is motivated by the unique risks associated with incorrect substance use predictions in clinical settings (Van Boekel et al., 2013; Dahl et al., 2022). While traditional metrics like accuracy or F1-score treat all errors equally, FPR specifically captures the rate of unwarranted "positive" classifications—a critical concern when dealing with sensitive patient information. High FPR values indicate that models frequently make unjustified drug use predictions, which could lead to:

- Patient stigmatization and potential discrimination
- Reduced quality of care due to biased provider perceptions
- Diminished trust in automated clinical decision support systems

Conversely, lower FPR values suggest better model reliability in avoiding these harmful misclassifications. While comprehensive evaluation would benefit from additional metrics, FPR serves as a particularly relevant indicator for assessing model safety and reliability in clinical applications.

D Model Fine-tuning and Computational Resources

We fine-tuned Llama-8B using LoRA with rank 64 and dropout 0.1. Key training parameters include a learning rate of 2e-4, batch size of 4, and 5 training epochs. Training was conducted on 2 NVIDIA A100 GPUs for approximately 3 hours using mixed precision (FP16). For our main experiments, we used several large language models: Llama-70B (70B parameters), Qwen-72B (72B parameters), Llama3-Med42-70B (70B parameters), and our fine-tuned Llama-8B (8B parameters). The inference experiments across all models required approximately 100 GPU hours on 2 NVIDIA A100 GPUs. This computational budget covered all experimental settings including zero-shot, in-context learning, and the evaluation of various mitigation strategies.

E Trigger Removal Experiments

Table 3: Impact of Trigger Removal on Llama 3.1 Models False Positive Rates (%)

	Llama 3.1 70b Zero-shot			Llama 3.1 8b SFT		
Cases	Full	Without Alcohol	Without Smoking	Full	Without Alcohol	Without Smoking
Alcohol-positive	66.21	55.17	64.14	32.41	26.90	33.10
Smoking-positive	61.11	54.94	56.79	36.42	32.10	31.48
Alcohol-negative	28.83	25.23	23.87	12.16	12.16	8.11
Smoking-negative	29.76	22.93	26.34	7.32	6.83	7.32
Smoking+Alcohol	73.26	65.12	72.09	40.70	32.56	41.86

Table 4: Impact of Trigger Removal on Additional Models' False Positive Rates (%)

	Llama 3.1 70B ICL			Llama3-Med42-70B			Qwen-72B		
Cases	Full	Without Alcohol	Without Smoking	Full	Without Alcohol	Without Smoking	Full	Without Alcohol	Without Smoking
Alcohol-positive	48.28	38.62	47.59	66.90	53.10	64.83	62.76	51.72	54.48
Smoking-positive	36.42	32.72	32.09	57.41	51.85	52.47	53.09	45.68	51.23
Alcohol-negative	11.71	16.22	10.81	16.22	16.22	13.96	46.85	45.05	47.75
Smoking-negative	18.05	14.15	15.12	19.51	14.15	19.51	53.17	49.27	49.76
Smoking+Alcohol	51.16	44.19	46.51	76.74	66.28	73.26	56.98	43.02	50.00

F Mitigation Experiments

Table 5: Impact of Mitigation Strategies on Additional Models' False Positive Rates (%)

	Llama3-Med42-70B				Qwen-72B			
Cases	ICL	CoT	Warning	Increased Examples	ICL	CoT	Warning	Increased Examples
Alcohol-positive	66.90	48.28	62.76	63.45	62.76	28.97	34.38	36.55
Smoking-positive	57.41	35.19	53.09	50.62	53.09	23.46	32.09	33.33
Alcohol-negative	16.22	6.76	16.67	15.76	46.85	19.82	22.07	26.12
Smoking-negative	19.51	13.66	18.54	18.05	53.17	17.07	25.85	29.27
Smoking+Alcohol	76.74	53.49	72.09	68.60	56.98	32.56	37.21	41.86

Enhancing NER by Harnessing Multiple Datasets with Conditional Variational Autoencoders

Taku Oi and **Makoto Miwa**
Toyota Technological Institute
Correspondence: makoto-miwa@toyota-ti.ac.jp

Abstract

We propose a novel method to integrate a Conditional Variational Autoencoder (CVAE) into a span-based Named Entity Recognition (NER) model. This approach models shared and unshared information among labels in multiple datasets, thereby easing training on these datasets. Experimental results using multiple biomedical datasets demonstrate the effectiveness of the proposed method, showing improved performance on the BioRED dataset. Our source code for this implementation is publicly available at GitHub[1].

1 Introduction

Named Entity Recognition (NER) is a fundamental Natural Language Processing (NLP) task, serving as a crucial first step in information extraction. Although Large Language Models (LLMs) with zero- or few-shot learning have been widely investigated, supervised learning or full fine-tuning remains essential for achieving high performance (Zhong and Chen, 2021; Luo et al., 2023; Wang et al., 2023; Munnangi et al., 2024; Zhou et al., 2024). The performance of such NER models depends on the quantity of labeled data, which is often costly to create manually.

One approach to increase labeled training data is to combine existing labeled datasets (Luo et al., 2022; Islamaj et al., 2021a). However, even datasets targeting the same types of named entities have different type definitions and annotation criteria, making it challenging to merge them into a unified training dataset. For instance, in the biomedical NER datasets NLMChem (Islamaj et al., 2021a) and BioRED (Luo et al., 2022), which both include a *Chemical* entity type, the term "hematoxylin" is labeled as *Chemical* in NLMChem, but not in BioRED. This discrepancy arises because BioRED's annotation guidelines explicitly exclude staining reagents from the *Chemical* label.

A common strategy for utilizing multiple datasets is multi-task learning (MTL) (Wang et al., 2018; Zuo and Zhang, 2020; Rodriguez et al., 2022). MTL shares a base model among tasks while maintaining a separate classification layer for each dataset, which allows for learning without explicitly addressing dataset-specific differences. However, MTL cannot account for relationships among labels in different datasets. Luo et al. (2023) tackled this issue by manually editing additional datasets to align with the target dataset, which improved performance, but the manual editing process is costly.

Span-based approaches are widely used for NER (Sohrab and Miwa, 2018; Ouchi et al., 2020; Nguyen et al., 2023) due to their simplicity and effectiveness. Span representations are critical for these methods, as they directly influence the ability to accurately identify and classify named entities. Nguyen et al. (2023) explored integrating Variational Autoencoders (VAE) to enhance span representations. Inspired by this, we employ a span-based approach with Conditional VAE (CVAE) to model spans using label-specific conditions that are both shared and unshared across multiple datasets.

This study proposes a method that models the relationships between labels in different datasets as label-specific conditions using CVAE, thereby incorporating these conditions to alleviate differences across datasets. The contributions of this paper are as follows:

- We propose a span-based NER method to integrate multiple existing datasets, despite their differing type definitions and annotation criteria.
- We propose integrating CVAE to incorporate and model the relationships between labels from multiple datasets.

[1] https://github.com/tti-kde/CVAE-NER

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1107–1117
July 27 - August 1, 2025

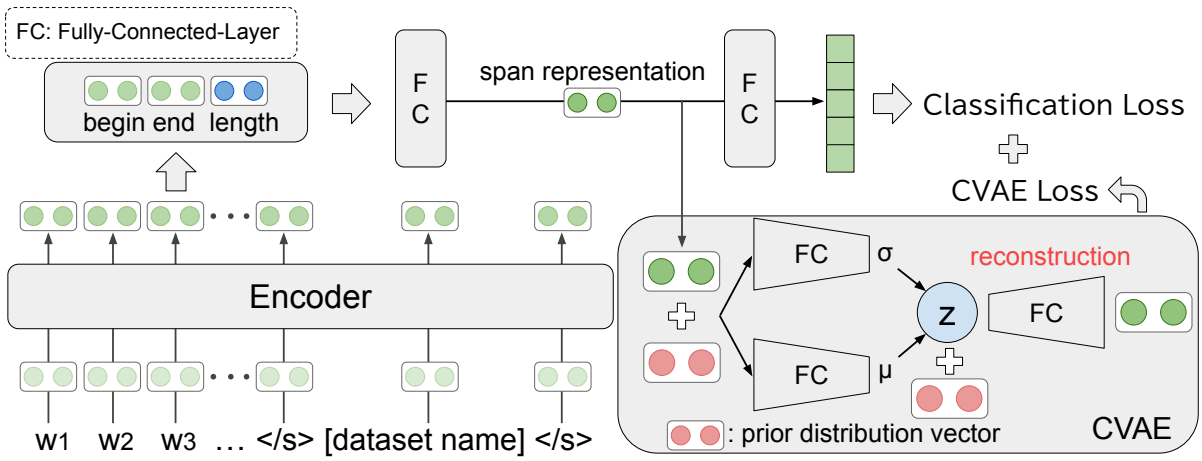


Figure 1: Overview of the proposed method, illustrating the integration of CVAE into a span-based NER model. Within the CVAE component, the prior distribution vector provides information regarding shared and unshared labels to the model.

- We demonstrate the effectiveness of our method through experiments on multiple biomedical datasets, achieving improved performance on the BioRED (Luo et al., 2023) dataset with either of the two encoders used.

2 Related Work

2.1 Span-based NER model

Span-based models, which directly classify specific text spans within a document, have recently gained attention (Zhong and Chen, 2021; Sohrab and Miwa, 2018) due to their simplicity and ability to model entity spans directly, unlike sequence labeling models that process tokens to represent entities. Zhong and Chen (2021) represent each target span by concatenating the embeddings of the first and last tokens from the encoder with the span length embedding. The span representations are then passed through a fully-connected layer, followed by a Softmax layer for classification. The model demonstrated high performance in NER and achieved state-of-the-art performance in end-to-end entity and relation extraction.

Span representations are crucial in span-based NER models. Ouchi et al. (2020) enhanced interpretability by learning the similarity between span representations and assessing this similarity during prediction. Nguyen et al. (2023) proposed a span-based NER model that incorporates span reconstruction and synonym generation using VAE. Span reconstruction ensures that instance-specific information is retained in the span representations, thereby improving extraction performance.

2.2 NER using multiple datasets

MTL (Rodriguez et al., 2022; Zuo and Zhang, 2020) employs a shared encoder and a separate classification layer for each dataset during training. Recent approaches leverage the capabilities of LLMs for MTL (e.g., Zhou et al. (2024)). However, these approaches do not explicitly account for shared label information across diverse datasets.

Luo et al. (2023) trained NER on multiple datasets targeting six types of entities: *Disease*, *Chemical*, *Gene*, *Species*, *Variant*, and *Cell line*, to improve performance on the BioRED dataset. Additional datasets underwent manual editing to ensure each contained only a single label type through label integration, deletion, and adjustment of label spans to align annotation criteria. The method outperformed existing MTL methods but required costly manual annotations. Our approach uniquely addresses the label differences through direct modeling with CVAE, which has not been explored in existing methods.

3 Method

This study proposes integrating CVAE into an NER model to incorporate a prior distribution that provides the model with information about shared and unshared labels among the datasets as a label-specific condition. We anticipate that this condition will capture relationships between labels in multiple datasets, enabling the effective use of existing labeled datasets. An overview of the proposed model is illustrated in Figure 1.

We follow Zhong and Chen (2021) to model the

representation of a span $x_1, \ldots, x_n$. The representation is constructed by concatenating the representations of the beginning token and end token from the encoder, along with the embeddings of the span length, as shown in Equation (1):

$$\boldsymbol{h}_{span} = \mathrm{Linear}(\mathrm{Concat}(\boldsymbol{x}_1, \boldsymbol{x}_n, \Phi(n))), \quad (1)$$

where $\mathrm{Linear}(\cdot)$ is a fully-connected layer and $\mathrm{Concat}(\cdot)$ is vector concatenation, $\boldsymbol{x}_i$ is the output representation of a token x_i from the encoder, and $\Phi(n)$ is the embedding of the span length n. The dataset name is added as a special token to the input text to ensure the model recognizes inputs from different datasets. The span representation is classified using two fully-connected layers, followed by a Softmax layer. Cross-entropy loss (L_{CE}) is employed as the loss function.

In CVAE, the span representation is concatenated with a prior distribution vector, which functions as the conditional parameter during the reconstruction phase. The prior distribution vector is derived from the correct label and encodes the shared and unshared label information across datasets[2]. After concatenating with the prior distribution vector, each span representation is processed through two fully-connected layers to estimate the mean (μ) and variance (σ) parameters of the latent distribution. A vector z, sampled from the estimated distribution, is further concatenated with the prior distribution vector and then processed through two fully-connected layers for reconstruction. The original loss function of CVAE L_{CVAE} (Kingma et al., 2014) is employed, which consists of the reconstruction error and the Kullback-Leibler (KL) divergence, as follows:

$$L_{CVAE} = -\log p(\boldsymbol{h}_{span}|z) + KL[q(z|\boldsymbol{h}_{span})||p(z)].$$

The prior distribution vector is prepared as trainable parameters. The vector is obtained by concatenating two one-hot vectors: one representing the labels across all datasets, with a "1" at the position corresponding to each label, and the other representing the corresponding (shared) label in the target dataset, including a negative label, with a "1" for each shared label.

As a simple example, consider the case where a target corpus tags Chemical$_T$ and Protein$_T$, and a source corpus tags Disease$_S$ and Protein$_S$, with Protein being a shared entity type across datasets. The first component of the prior distribution vector represents the labels across all datasets. For this example, this component would form a 6-dimensional space including negative labels: [O$_T$, Chemical$_T$, Protein$_T$, O$_S$, Disease$_S$, Protein$_S$]. The second component represents the corresponding (shared) label in the target dataset including a negative label. For this example, this component would form a 3-dimensional space: [O, Chemical$_T$, Protein$_T$]. Consequently, the entire prior distribution vector is expressed within a combined 9-dimensional space. For instance, for an entity labeled Protein$_S$ from the source corpus, its prior distribution vector would be [0, 0, 0, 0, 0, 1, 0, 0, 1]. Here, the "1"s are placed at the dimension corresponding to Protein$_S$ in the corpus-specific space and at the dimension corresponding to the shared Protein$_T$ label in the shared target space.

Finally, the weighted sum of the two losses is used as the entire loss L for training:

$$L = L_{CE} + \alpha L_{CVAE}.$$

It is important to note that CVAE is not used during inference since it requires correct labels. Instead, the trained classifier alone is used to recognize named entities. Our method thus requires no additional computation during inference, making it suitable for practical applications.

4 Experimental Setup

We describe the datasets and models used in the experiments. Detailed settings and hyperparameters are provided in Appendix A.

4.1 Datasets

We employed 10 labeled biomedical datasets following Luo et al. (2023). The number of documents and the named entities labeled in these datasets are shown in Table 6 of Appendix B. We used the development set of BioRED for evaluation during training, while the development sets of other datasets were used as training data. Unlike Luo et al. (2023), which edited the spans and target labels of entities in additional datasets to follow the BioRED standards, we utilized the original datasets without any such edits.

4.2 Prior distribution vector

As explained in §3, the prior distribution vector provided as a condition to CVAE is constructed

[2]The actual settings for the prior distribution vector will be shown in §4.2.

Model	Encoder	All	*Disease*	*Chemical*	*Gene*	*Variant*	*Species*	*Cell line*
Luo et al. (2023)[3]	BERT	91.26	88.07	90.98	92.40	88.51	97.50	90.53
Single		87.86	82.33	89.42	89.63	81.38	96.53	80.90
Multi		93.22	90.66	**93.41**	95.18	86.84	97.96	79.55
Single + CVAE		87.28	81.93	89.76	89.64	77.80	94.74	75.00
Multi + CVAE		94.19	91.88	92.98	**96.47**	92.80	97.48	81.82
Single	T5	89.89	86.52	87.99	91.19	90.56	97.73	86.67
Multi		93.99	92.82	92.16	95.89	90.07	98.00	82.76
Single + CVAE		89.87	85.81	89.45	90.72	89.18	97.62	**91.84**
Multi + CVAE		**94.36**	**92.84**	92.89	95.77	**94.29**	**98.23**	78.57

Table 1: F1 scores [%] on the BioRED test set, serving as the primary metric. The 'Encoder' column specifies T5-3B for 'T5' and PubMedBERT for 'BERT'. The highest F1 score for each entity type is highlightted in bold.

by concatenating two one-hot vectors. The first vector is a one-hot vector with a dimension of 47, representing all labels listed in Table 6, including a negative label for each dataset. The second vector is a seven-dimensional one-hot vector (six types + negative) representing the shared labels based on the BioRED labels. We use the mapping of Luo et al. (2023), which mapped the labels from the additional datasets to one of the BioRED labels, including the negative label (e.g., treating the Gene and FamilyName labels from NLMGene as equivalent to the Gene label in BioRED). The details of this correspondence are shown in Appendix C.

4.3 Comparison settings

We employed the encoder part of T5 (Text-To-Text Transfer Transformer)-3B (Raffel et al., 2020) as the encoder of our model. We also employed PubMedBERT (Gu et al., 2021) as the encoder for a fair comparison with Luo et al. (2023). We employ the F1 score as our primary metric. We compared the following settings:

- **Single**
  The span-based NER model trained on a single target dataset.

- **Multi**
  The span-based NER model trained on multiple datasets in a pure MTL setting, where separate classification layers are prepared for each dataset while the base encoder layer is shared.

- **Multi+CVAE**
  The span-based NER model with CVAE that incorporates the prior distribution vector described in §4.2.

- **Multi+CVAE (Fixed)**
  The span-based NER model with CVAE that fixes the prior distribution vector after initialization.

- **Multi+CVAE (Unshared)**
  The span-based NER model with CVAE that initializes the prior distribution vector using only the first one-hot vector (labels in each dataset) in §4.2.

- **Multi+CVAE (Shared label)**
  The span-based NER model with CVAE that initializes the prior distribution vector using only the second one-hot vector (shared label information) in §4.2.

- **Multi+CVAE (Random)**
  The span-based NER model with CVAE that initializes the prior distribution vector randomly.

- **Multi (Shared label)**
  The span-based NER model trained on multiple datasets without CVAE, where shared labels across datasets are assigned to the same output layer.

5 Results

The overall performance is summarized in Table 1. Compared with the state-of-the-art (SOTA) model by Luo et al. (2023), our model demonstrates enhanced performance across all label types when using multiple datasets, with the exception of *Cell line*. This *Cell line* result can be attributed to the limited number of instances in the test set, as shown in Table 5 of Appendix B, which makes

[3]The score is taken from the original paper.

Model	All
Multi+CVAE	**94.36**
Multi+CVAE (Fixed)	93.99
Multi+CVAE (Unshared)	94.12
Multi+CVAE (Shared label)	94.20
Multi+CVAE (Random)	94.07
Multi (Shared label)	92.52

Table 2: Ablation study using the T5 encoder.

the evaluation sensitive to variations of individual cases. When trained on a single dataset with CVAE (Single+CVAE), the performance slightly degraded when the BERT encoder was employed. This may be because the representations of different labels are not sufficiently distant due to the interaction between labels caused by the CVAE loss. As for the performance of our models with multiple datasets, both the Multi and Multi+CVAE models achieved better F1 scores regardless of the encoder, compared to the corresponding Single model trained solely on BioRED. Furthermore, the CVAE model improved the F1 score for both encoders, suggesting its potential effectiveness in leveraging CVAE during training. This result indicates that the proposed model could capture some of the shared and unshared information among the datasets. This is further supported by the visualizations in Figure 4 of Appendix D, where instances with shared labels from different datasets have closer embeddings.

The ablation study with modified conditions for the prior distribution vector is shown in Table 2. All settings showed inferior performance compared to the proposed model, demonstrating the effectiveness of using the shared labels and tuning the prior distribution vector during training.

6 Conclusion

This study aimed to alleviate the differences in labeling criteria among different datasets to increase the amount of training data. To achieve this, we proposed a method that incorporates a CVAE-based loss function into a span-based NER model, which considers the differences in labeling criteria without the need for manual edits. Our experimental results showed that our CVAE-based approach can leverage multiple datasets while accommodating their labeling discrepancies for either of the two encoders used.

For future work, we will investigate a fully automated method by making the prior distribution vector independent from a specific dataset and eliminating the need for manual definition of the shared vectors, aiming to develop a model that can improve performance on additional datasets as well and be applied to other datasets. Additionally, to further demonstrate broader generalizability, we will evaluate CVAE with other backbone NER models.

Acknowledgments

The authors gratefully acknowledge the assistance provided by AI tools, specifically GitHub Copilot and Claude 3.5 Sonnet, in supporting our programming and writing processes.

Limitations

Our model is based on the approach proposed by Zhong and Chen (2021), where the maximum span length is set to 10. As a result, our model cannot extract named entities that exceed this length constraint. Additionally, as our model primarily utilizes an encoder-based architecture, it may not be straightforward to adapt it to LLMs. Although our method does not require manual annotation, it still requires manual effort to define the prior distribution vector. Due to this limitation, we have evaluated our method only on the BioRED dataset, and the application to other datasets has not been investigated.

Ethical Considerations

This paper utilizes the pre-trained LLM T5 as the base model, which may contain inherent biases due to its pre-training processes. However, we employ only the encoder part of T5 and fine-tune the model for NER, so such potential biases may be mitigated. Furthermore, we use public datasets only for scientific research purposes.

References

Cecilia Arighi, Lynette Hirschman, Thomas Lemberger, et al. 2017. Bio-id track overview. In *BioCreative VI Workshop*, pages 28–31, Bethesda, MD, USA. BioCreative.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10.

Martin Gerner, Goran Nenadic, and Casey M Bergman. 2010. Linnaeus: a species name identification sys-

tem for biomedical literature. *BMC bioinformatics*, 11(1):1–17.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, 3(1).

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Rezarta Islamaj, Robert Leaman, Sun Kim, Dongseop Kwon, Chih-Hsuan Wei, Donald C Comeau, Yifan Peng, David Cissel, Cathleen Coss, Carol Fisher, et al. 2021a. Nlm-chem, a new resource for chemical entity recognition in pubmed full text literature. *Scientific data*, 8(1):91.

Rezarta Islamaj, Chih-Hsuan Wei, David Cissel, Nicholas Miliaras, Olga Printseva, Oleg Rodionov, Keiko Sekiya, Janice Ward, and Zhiyong Lu. 2021b. Nlm-gene, a richly annotated gold standard dataset for gene entities that addresses ambiguity and multi-species gene recognition. *Journal of biomedical informatics*, 118:103779.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016:baw068.

Ling Luo, Po-Ting Lai, Chih-Hsuan Wei, Cecilia N Arighi, and Zhiyong Lu. 2022. Biored: a rich biomedical relation extraction dataset. *Briefings in Bioinformatics*, 23(5):bbac282.

Ling Luo, Chih-Hsuan Wei, Po-Ting Lai, Robert Leaman, Qingyu Chen, and Zhiyong Lu. 2023. AIONER: all-in-one scheme-based biomedical named entity recognition using deep learning. *Bioinformatics*, 39(5):btad310.

Monica Munnangi, Sergey Feldman, Byron Wallace, Silvio Amir, Tom Hope, and Aakanksha Naik. 2024. On-the-fly definition augmentation of LLMs for biomedical NER. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3833–3854, Mexico City, Mexico. Association for Computational Linguistics.

Nhung T. H. Nguyen, Makoto Miwa, and Sophia Ananiadou. 2023. Span-based named entity recognition by generating and compressing information. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1984–1996, Dubrovnik, Croatia. Association for Computational Linguistics.

Hiroki Ouchi, Jun Suzuki, Sosuke Kobayashi, Sho Yokoi, Tatsuki Kuribayashi, Ryuto Konno, and Kentaro Inui. 2020. Instance-based learning of span representations: A case study through named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6452–6459, Online. Association for Computational Linguistics.

Evangelos Pafilis, Sune P Frankild, Lucia Fanini, Sarah Faulwetter, Christina Pavloudi, Aikaterini Vasileiadou, Christos Arvanitidis, and Lars Juhl Jensen. 2013. The species and organisms resources for fast and accurate identification of taxonomic names in text. *PloS one*, 8(6):e65390.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nicholas E. Rodriguez, Mai Nguyen, and Bridget T. McInnes. 2022. Effects of data and entity ablation on multitask learning models for biomedical entity recognition. *Journal of Biomedical Informatics*, 130:104062.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. Instructuie: Multi-task instruction tuning for unified information extraction. *Preprint*, arXiv:2304.08085.

Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, Curtis Langlotz, and Jiawei Han. 2018. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, 35(10):1745–1752.

Chih-Hsuan Wei, Alexis Allot, Kevin Riehle, Aleksandar Milosavljevic, and Zhiyong Lu. 2022. tmvar 3.0: an improved variant concept recognition and normalization tool. *Bioinformatics*, 38(18):4449–4451.

Chih-Hsuan Wei, Hung-Yu Kao, Zhiyong Lu, et al. 2015. Gnormplus: an integrative approach for tagging genes, gene families, and protein domains. *BioMed research international*, 2015.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. UniversalNER: Targeted distillation from large language models for open named entity recognition. In *The Twelfth International Conference on Learning Representations*.

Mei Zuo and Yang Zhang. 2020. Dataset-aware multi-task learning approaches for biomedical named entity recognition. *Bioinformatics*, 36(15):4331–4338.

A Detailed training settings

We listed the number of parameters for the encoder models used in the experiments in Table 3. In experiments using the T5 encoder, we employed Low-Rank Adaptation (LoRA) (Hu et al., 2022) and model parameter quantization during training. In addition, the hyperparameters used in the experiments are summarized in Table 4. In the experiments, we took 16 examples from each dataset. We then performed training with a mini-batch size of 64, effectively including examples from multiple datasets by weight accumulation and combining losses from four datasets before updating the model. The weight of the CVAE loss (i.e., α) was selected based on the results from Figure 2, which were obtained from experiments using the T5-base model. To ensure optimal training, the model was evaluated on the development set after each epoch, and training was terminated if no improvement in validation performance was observed for five consecutive epochs. All experiments were conducted on a computing infrastructure equipped with four NVIDIA V100 GPUs. All experimental results reported in this paper are based on a single run with a fixed random seed.

B Dataset statistics

Table 5 presents the statistics of entity counts in the train, development, and test sets of the BioRED dataset. The statistics of the datasets and the list of named entity labels are summarized in Table 6. Note that all datasets used in this study are in English.

C Correspondence of labels

The correspondence of the labels of each dataset to the BioRED labels is summarized in Table 7. This correspondence is from Luo et al. (2023). For spans with no correspondence to the BioRED labels, the second one-hot vectors of their prior distribution vectors have no active values.

We visualize the prior distribution vectors after training using t-SNE in Figure 3. The visualization demonstrates that the labels in Table 7 remain clustered around their corresponding BioRED labels, and labels with no correspondence remain far from the BioRED labels. This suggests that the initialization information is mostly preserved throughout the training process.

Model	#Params
T5-3B	876,802,048
T5-base	86,043,264
PubMedBERT	109,482,240

Table 3: Numbers of parameters (#Params) for the encoder models used in the experiments.

Parameter	Value
Learning rate	7e-4
α	1e-4
Mini-batch size	64
CVAE hidden dim	150
Max span length	10
Span length embedding dim	150
LoRA rank	32

Table 4: Hyperparameter settings

	Train	Dev	Test
Document	400	100	100
All	13,351	3,533	3,535
Disease	3,646	982	917
Chemical	2,853	822	754
Gene	4,430	1,087	1,180
Variant	890	250	241
Species	1,429	370	393
Cell line	103	22	50

Table 5: Detailed statistics of entity counts across train, development, and test sets of the BioRED dataset (Luo et al., 2022), categorized by the entity types.

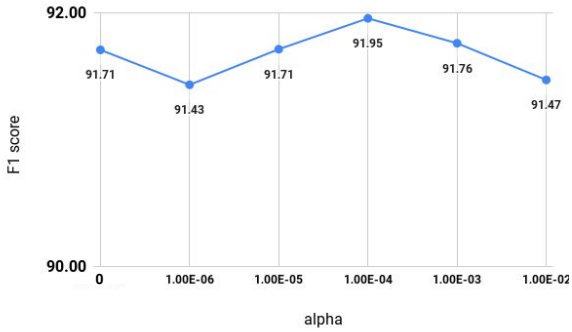


Figure 2: F1 scores [%] on the BioRED development set, plotted against the weight α applied to the CVAE loss, using the T5-base encoder; $\alpha = 0$ indicates the F1 score obtained without the CVAE loss.

D Visualization of instance embeddings

Figure 4 visualizes the embeddings of instances in the development set of the BioRED dataset and the test sets of the nine additional datasets using

Datasets	Size	Labels
BioRED (Luo et al., 2022)	600	*Disease*, *Chemical*, *Gene*, *Variant*, *Species*, *Cell line*
BC5CDR (Li et al., 2016)	1500	*Disease*, *Chemical*
BioID (Arighi et al., 2017)	570 full	*Cell line*
GNormPlus (Wei et al., 2015)	694	*FamilyName*, *Gene*, *DomainMotif*
Linnaeus (Gerner et al., 2010)	100 full	*Species*
NCBIdisease (Doğan et al., 2014)	793	*DiseaseClass*, *SpecificDisease*, *CompositeMention*, *Modifier*
NLMChem (Islamaj et al., 2021a)	150 full	*Chemical*, *NonStandardRef*, *OTHER*
NLMGene (Islamaj et al., 2021b)	550	*Gene*, *FamilyName*, *Cell*, *DomainMotif*, *ChromosomeLocation*
SPECIES800 (Pafilis et al., 2013)	800	*Species*
tmVar3 (Wei et al., 2022)	500	*Gene*, *Species*, *Disease*, *DNAMutation*, *ProteinMutation*, *OtherMutation*, *Cell line*, *AcidChange*, *SNP*, *DNAAllele*, *ProteinAllele*

Table 6: Summary of the datasets used. Datasets with "full" in the size column contain full-text data, while the others contain only abstracts.

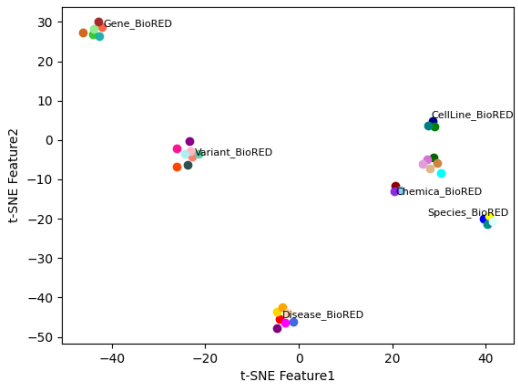


Figure 3: Visualization of prior distribution vectors after training by t-SNE.

t-SNE (van der Maaten and Hinton, 2008). The instances are taken from 300 sentences for each dataset. Compared to the two figures on the left side by the Multi model, the two on the right side using the proposed Multi+CVAE model show a tendency for *Disease* and *Species* instances from different datasets to have closer embeddings. This tendency is particularly pronounced for *Species*, which may contribute to improved performance for *Species* in BioRED.

E Evaluation on the additional datasets

We evaluated both Multi and Multi+CVAE models on nine additional datasets, with results shown in Appendix E. Our proposed method improved performance over the Multi baseline on only two of the nine datasets. As discussed in the Conclusion, this limited improvement is attributed to using prior distribution vectors specifically designed for BioRED, highlighting an area that requires future investigation.

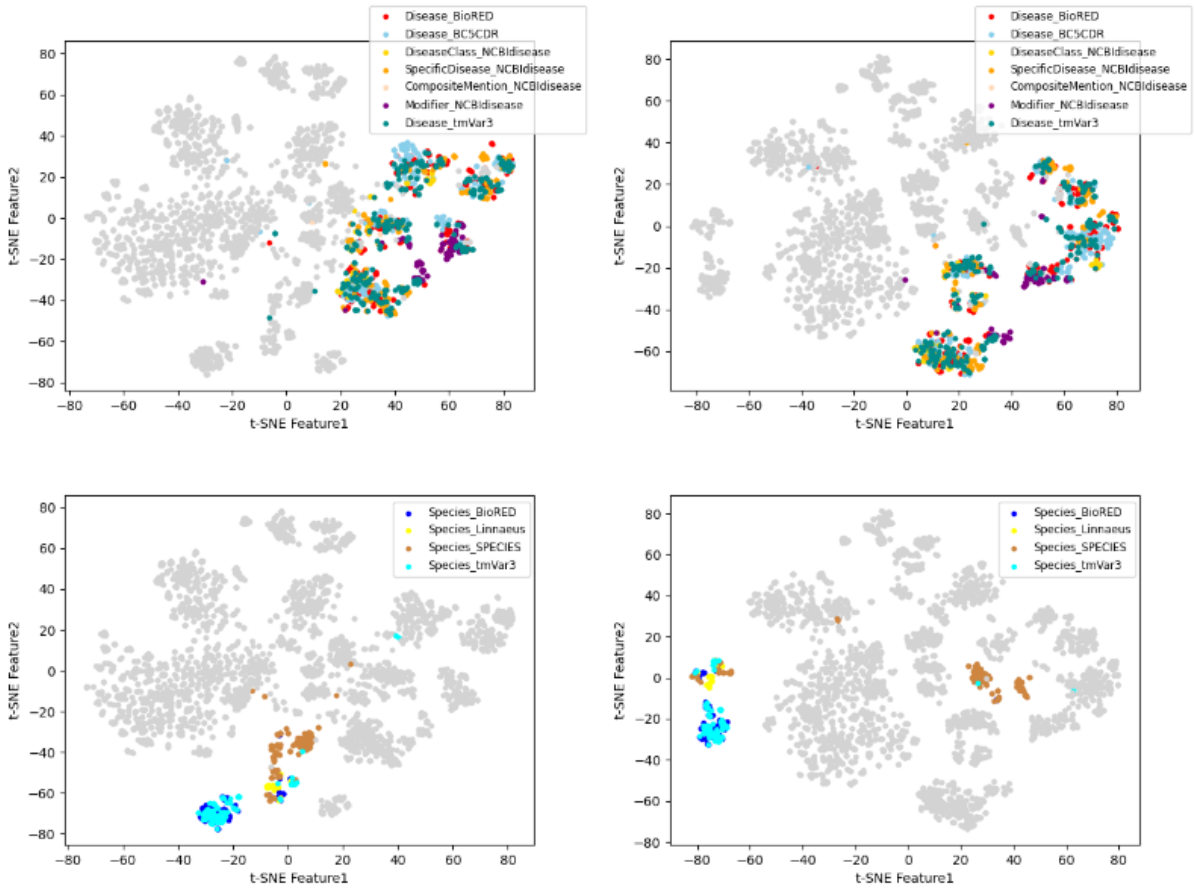


Figure 4: t-SNE Visualization of instance embeddings from the BioRED development set and additional datasets. Upper left: *Disease* Multi+CVAE, upper right: *Disease* Multi, lower left: *Species* Multi+CVAE, lower right: *Species* Multi

Dataset	Label	Corresponding label in BioRED
BC5CDR (Li et al., 2016)	*Disease*	*Disease*
	Chemical	*Chemical*
BioID (Arighi et al., 2017)	*Cell line*	*Cell line*
GNormPlus (Wei et al., 2015)	*Gene*	*Gene*
	FamilyName	*Gene*
Linnaeus (Gerner et al., 2010)	*Species*	*Species*
NCBIdisease (Doğan et al., 2014)	*DiseaseClass*	*Disease*
	SpecificDisease	*Disease*
	CompositeMention	*Disease*
	Modifier	*Disease*
NLMchem (Islamaj et al., 2021a)	*Chemical*	*Chemical*
NLMGene (Islamaj et al., 2021b)	*Gene*	*Gene*
	FamilyName	*Gene*
SPECIES800 (Pafilis et al., 2013)	*Species*	*Species*
tmVar3 (Wei et al., 2022)	*DNAMutation*	*Variant*
	ProteinMutation	*Variant*
	OtherMutation	*Variant*
	AcidChange	*Variant*
	SNP	*Variant*
	DNAAllele	*Variant*
	ProteinAllele	*Variant*
	Disease	*Disease*
	Chemical	*Chemical*
	Gene	*Gene*
	Species	*Species*
	Cell line	*Cell line*

Table 7: Correspondence of labels of the additional datasets to BioRED labels. The labels *DomainMotif* in GNormplus, *NonStandardRef* and *OTHER* in NLMChem, and *Cell*, *DomainMotif*, and *ChromosomeLocation* in NLMGene have no correspondence to the BioRED labels.

Dataset	Multi	Multi+CVAE
BC5CDR (Li et al., 2016)	91.42	**91.59**
BioID (Arighi et al., 2017)	**90.71**	90.01
GNormPlus (Wei et al., 2015)	**80.97**	80.68
Linnaeus (Gerner et al., 2010)	**94.69**	93.84
NCBIdisease (Doğan et al., 2014)	**85.77**	84.11
NLMChem (Islamaj et al., 2021a)	**82.74**	82.12
NLMGene (Islamaj et al., 2021b)	**85.12**	84.76
SPECIES800 (Pafilis et al., 2013)	**81.12**	77.64
tmVar3 (Wei et al., 2022)	85.29	**91.87**

Table 8: F1 scores [%] on test data of the additional datasets.

CHEER-Ekman: Fine-grained Embodied Emotion Classification

Phan Anh Duong, Cat Luong, Divyesh Bommana, Tianyu Jiang
University of Cincinnati
{duongap, luongcn, bommandh}@mail.uc.edu, tianyu.jiang@uc.edu

Abstract

Emotions manifest through physical experiences and bodily reactions, yet identifying such embodied emotions in text remains understudied. We present an embodied emotion classification dataset, CHEER-Ekman,[1] extending the existing binary embodied emotion dataset with Ekman's six basic emotion categories. Using automatic best-worst scaling with large language models, we achieve performance superior to supervised approaches on our new dataset. Our investigation reveals that simplified prompting instructions and chain-of-thought reasoning significantly improve emotion recognition accuracy, enabling smaller models to achieve competitive performance with larger ones.

1 Introduction

Emotions are not merely abstract mental states; they are deeply intertwined with somatic experiences. When we feel joy, our faces light up with smiles; when we are scared, our hearts race and our hands tremble. These physical reactions are more than just side effects—they are part of how we experience and express emotions. This concept, known as embodied emotion, suggests that our bodies play a key role in how we feel, perceive, and understand emotions (Lakoff and Johnson, 1999; Niedenthal, 2007). In natural language, these connections surface as descriptions of physiological reactions (e.g., "my stomach churned in disgust") or unintentional physical actions (e.g., "she stomped her feet in frustration")—phenomena termed *embodied emotions*. Recognizing such expressions is pivotal for understanding implicit emotional cues in narratives. While recent advances in NLP have focused more on explicit emotion classification (Mohammad et al., 2018) or sentiment analysis (Rosenthal et al., 2017), the subtler task of identifying embodied emotions remains less explored, despite its psychological grounding and practical relevance.

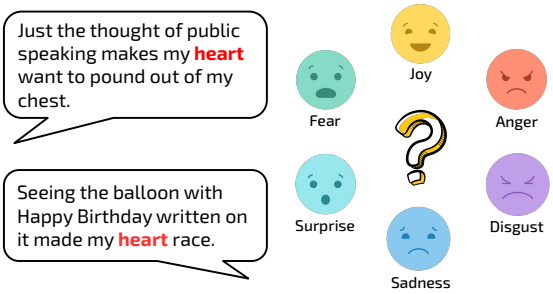


Figure 1: Illustration of embodied emotions classified into six categories.

The CHEER dataset (Zhuang et al., 2024) filled a gap in this field by providing a collection of sentences where body parts are used to express emotions. The dataset includes 7,300 human-annotated sentences containing body part references, proposing a binary classification task which we will refer to as "embodied emotion detection." However, one limitation of this work is that it does not distinguish between different types of emotions—for instance, whether a racing heart signals fear or excitement. The framework of Ekman's (1992) basic emotions offers a potential solution to this limitation. By linking embodied expressions to these specific emotions, we can build systems that better understand human emotional experiences.

To achieve this goal, we extend the CHEER dataset by annotating all its 1,350 positive samples with six Ekman emotion labels (*Joy*, *Sadness*, *Anger*, *Disgust*, *Fear*, and *Surprise*), creating a new dataset, **CHEER-Ekman**, as illustrated in Figure 1. For clarity, we refer to the novel classification task produced by this dataset as "embodied emotion *classification*," as compared to the binary task discussed above. We adopt the automatic best-worst scaling (BWS) technique (Kiritchenko and Mohammad, 2017; Bagdon et al., 2024) with large

[1] https://github.com/menamerai/cheer-ekman

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1118–1131
July 27 - August 1, 2025

language models (LLMs) to tackle the task. Our experiments show that using Llama 3.1 8B with BWS significantly outperforms zero-shot prompting. The best BWS experiment achieved 50.6 F1-score, surpassing supervised BERT (49.6) and beating zero-shot approaches by around 20 points. Building on Zhuang et al.'s (2024) investigation of LLMs' capability for embodied emotion detection, we further explore prompting techniques that enhanced the detection task. Our experimental results reveal that LLMs can make better recognition when instructions are rephrased in plain, easily understood language, which boosts F1 by nearly 30 points compared to technical definitions. Moreover, chain-of-thought reasoning enables an 8B parameter model to nearly match a 70B model, closing the performance gap to within 7 F1 points. In summary, our contributions are three-fold:

1. We present CHEER-Ekman, an extension of the CHEER dataset that enriches embodied emotion expressions with fine-grained Ekman emotion labels, addressing a critical gap in understanding how specific emotions manifest through bodily expressions. Our dataset is available at: `https://github.com/menamerai/cheer-ekman`.

2. We demonstrate that the automatic best-worst scaling technique enables LLMs to perform emotion classification without any task-specific training, achieving performance that exceeds supervised approaches.

3. We reveal that counterintuitively, simplified everyday language in prompts dramatically outperforms technical definitions for embodied emotion tasks, and that structured reasoning through chain-of-thought can allow smaller language models to perform at a level closer to larger language models on our task.

2 Related Work

Emotion recognition in natural language processing has been extensively studied, with researchers focusing more on explicit emotion using datasets like SemEval (Strapparava and Mihalcea, 2007; Mohammad et al., 2018) and GoEmotions (Demszky et al., 2020). Recent research has expanded to explore nuanced aspects of emotion expression (Li et al., 2021), including emotion intensity prediction (Mohammad, 2018; Bagdon et al., 2024) and the detection of subtle emotional cues in dialogues (Poria et al., 2019; Ghosal et al., 2020; Li et al., 2022). The advent of large language models (LLMs) has catalyzed significant advances in emotion understanding capabilities (Lee et al., 2024; Sabour et al., 2024; Zhao et al., 2024; Liu et al., 2024).

While these works contribute to a deeper understanding of emotion detection in text, the embodied nature of emotions—how physical sensations and actions encode affective states—has received comparatively less attention. The concept of embodied emotion is rooted in cognitive science, particularly in the works of Lakoff and Johnson (1999) and Niedenthal (2007), which suggest that emotional experiences are closely tied to bodily states and actions. Despite the psychological grounding of embodied emotions, computational approaches to capturing them in text remain limited. Zhuang et al. (2024) introduced the CHEER dataset, which provides a collection of sentences where body parts are explicitly used to express emotions. Our work extends theirs by incorporating Ekman's six basic emotions into embodied emotion recognition, offering a more fine-grained classification system.

Another relevant line of work is the study of emotion taxonomy. Although recent advances in psychology have offered newer granular categories of emotions such as 27 emotions by Cowen and Keltner (2017), which has been adopted in both textual emotion datasets (Demszky et al., 2020) and visual emotion dataset (Kosti et al., 2019), we follow the vast majority of existing emotion datasets (Strapparava and Mihalcea, 2007; Mohammad et al., 2018; Poria et al., 2019) by utilizing the six basic emotions (*Joy*, *Sadness*, *Anger*, *Disgust*, *Fear*, and *Surprise*) proposed by Ekman (1992), which remain foundational due to their universality and simplicity. Future research may explore integrating alternative taxonomies into embodied emotion classification to enhance both granularity and coverage.

3 Methods

Our methodological approach comprises three key components that build upon and extend the work of Zhuang et al. (2024). First, we explore prompting strategies to enhance LLMs' capability to detect embodied emotions. Second, we introduce CHEER-Ekman, a refinement of the original CHEER dataset that adds fine-grained emotion labels. Finally, we adopt the BWS framework for

emotion classification that leverages comparative judgments to improve classification accuracy.

3.1 Prompting LLMs for Embodied Emotion Detection

To address the gap in prompt design within the Zhuang et al.'s (2024) framework, we first sought to enhance the embodied emotion detection task using state-of-the-art LLMs and explore the impact of prompt engineering on performance. Our approach centers on two key strategies: prompt simplification to mitigate linguistic complexity and chain-of-thought (CoT) prompting.

Prompt Simplification. We investigated the effects of linguistic and domain complexity by conducting experiments with the base Llama-3.1 (Grattafiori et al., 2024) and the recently released DeepSeek-R1 distilled version (DeepSeek-AI et al., 2025). Specifically, we compared two prompts: the base prompt used in Zhuang et al. (2024) and a simplified prompt, which reduces syntactic and lexical complexity to minimize potential comprehension barriers for LLMs.

Chain-of-Thought Prompting. We further explored eliciting reasoning from the model by implementing chain-of-thought (CoT) prompting. Based on Zhuang et al.'s (2024) annotation criteria for embodied emotion detection, we developed three CoT variants: a 2-step variant that evaluates emotional causation and purposeless expression, a 3-step variant that adds body part identification, and a simplified 2-step variant with reduced linguistic complexity. These variants allowed us to examine how explicit causal reasoning affects both the model's emotion detection performance and its understanding of body-emotion relationships.

3.2 Dataset Creation

While embodied emotion detection identifies emotional expressions through bodily movements, understanding the specific emotions conveyed requires more fine-grained annotation. To address this need, we propose **CHEER-Ekman**, a refined dataset extending the original CHEER corpus (Zhuang et al., 2024) by annotating its 1,350 positive embodied emotion instances with Ekman's (1992) six basic emotions (*Joy*, *Sadness*, *Anger*, *Disgust*, *Fear*, and *Surprise*). Our adoption of Ekman's basic emotions taxonomy balances granularity with practical considerations, as recent research by Liu et al. (2024) demonstrates that finer-grained emotion taxonomies often face sparsity issues, even in bigger datasets like GoEmotions (Demszky et al., 2020).

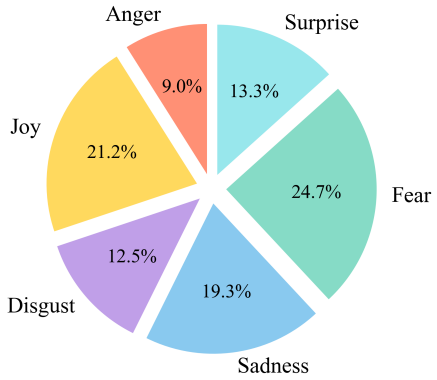


Figure 2: CHEER-Ekman dataset distribution of emotions.

Emotion	Sentence
Joy	... watched the fireflies with a loving look on his **face**.
Sadness	... frowning and scuffing his **feet** along the floor.
Fear	Marty nervously runs his **fingers** through his hair...
Anger	... makes me want to hit my **head** against the wall.
Disgust	Dean snorted incredulously, shaking his **head** in disbelief.
Surprise	... my **eyes** almost fell out of my head.

Table 1: Examples in our CHEER-Ekman datasets.

This approach also maintains consistency with Zhuang et al.'s (2024) methodology, which utilized emotion-associated adverbs derived from these basic emotions for weak supervision. We have also elected not to include the weakly labeled positive samples from the CHEER dataset, prioritizing our final dataset quality and reliability over quantity.

We recruited two annotators to label the 1,350 embodied emotion sentences from the original CHEER dataset. For each sentence, we provide the annotators with the sentence, the relevant body part, and up to three preceding sentences for context. We then ask the annotators to select one of the six emotions that best match the physical experience described through the body part. The pairwise inter-annotator agreement by Cohen's Kappa is 0.64, indicating good agreement. Finally, the annotators adjudicated their disagreements to produce the final gold labels. We show some example sentences with their annotated emotions in Table 1.

Figure 2 illustrates the emotion distribution of sentences in our newly constructed CHEER-Ekman dataset. Specifically, *Fear* is the most prevalent emotion (24.7%), followed by *Joy* (21.2%), *Sadness* (19.3%), *Surprise* (13.3%), and *Disgust* (12.5%). *Anger* appeared last at 9.0%.

3.3 BWS for Emotion Classification

To address the fine-grained emotion classification task proposed with CHEER-Ekman, we first tested zero-shot LLM prompting. However, the model often fails to adhere to the instructions and is prone to erroneous behaviors, leading to incorrect outputs. Inspired by recent work on automatic emotion intensity annotation using LLMs (Bagdon et al., 2024), we adopted the same best-worst scaling technique. The methodology involves presenting the LLM with tuples of four different sentences, instructing it to identify the body part instances that most and least represent a specific Ekman emotion. Then, the equation $\frac{\#Best - \#Worst}{\#Total}$ is used to calculate a score per sentence per emotion. Finally, we choose the emotion that receives the highest score to be the prediction for the sentence.

To examine how the number of comparisons affects classification accuracy, we tested a broader range of tuple counts, from $2N$ to $72N$, increasing by 50% at each step of expansion (where N is the number of instances to be classified). While Bagdon et al. (2024) found that more tuples improved accuracy in their experiments up to 12N, we expanded this investigation to 72N to further explore performance gain behaviors from scaling comparative rounds.

4 Evaluation

We conduct experiments to tackle both the embodied emotion detection and emotion classification tasks. The embodied emotion detection CHEER dataset contains 7,300 sentences, and our CHEER-Ekman dataset contains 1,350 sentences. We explored various strategies and models, and reported their F1-scores on both datasets.

4.1 Embodied Emotion Detection

Simple Prompting Analysis. To obtain the binary classification results, we directly compare the logit probabilities of "*True*" and "*False*" tokens instead of using text generation. This approach ensures deterministic outputs by avoiding the randomness inherent in sampling-based decoding methods, while also preventing potential output format violations that can occur during free-form generation. Table 2 shows that simplified prompts led to substantial performance improvements for the 70B parameter models. The F1-score increased by 29.5 points for Llama-3.1-70B, and by 41.6 points for DeepSeek-R1-70B (distilled on Llama), surpassing GPT 3.5 results reported in Zhuang et al. (2024).

Model	Macro F1	EE			Neutral		
		Pre	Rec	F1	Pre	Rec	F1
$\text{Llama}_{\text{base}}$	37.2	21.5	99.6	35.3	99.6	24.2	39.0
$\text{Llama}_{\text{simple}}$	66.7	37.5	89.3	52.8	96.9	68.9	80.6
$\text{DeepSeek}_{\text{base}}$	32.6	20.3	99.4	33.7	99.3	18.7	31.5
$\text{DeepSeek}_{\text{simple}}$	74.2	51.2	69.3	58.9	93.1	86.2	89.5
$\text{GPT 3.5}_{\text{base}}$	70.2	44.0	68.3	53.5	92.5	81.9	86.9
BERT	83.5	73.2	72.1	72.6	94.2	94.5	94.4

Table 2: Results comparison for Embodied Emotion Detection. Llama: Llama-3.1-70B. DeepSeek: DeepSeek-R1-Distilled-Llama-70B. GPT 3.5 and fine-tuned BERT numbers are from Zhuang et al. (2024). The *base* and *simple* subscripts indicate the type of prompts, which can be found in Table 5.

Model	Macro F1	EE			Neutral		
		Pre	Rec	F1	Pre	Rec	F1
$\text{Llama}_{\text{2-step}}$	53.4	26.2	80.8	39.6	93.0	52.7	67.2
$\text{Llama}_{\text{3-step}}$	54.8	24.9	53.4	34.0	87.3	66.5	75.5
$\text{Llama}_{\text{2-step-simple}}$	60.1	31.5	44.5	36.9	87.4	79.8	83.4
$\text{DeepSeek}_{\text{2-step}}$	52.2	26.4	90.8	40.9	96.1	47.3	63.4
$\text{DeepSeek}_{\text{3-step}}$	57.4	27.9	62.0	38.5	89.4	66.7	76.4
$\text{DeepSeek}_{\text{2-step-simple}}$	67.5	40.1	65.2	49.7	91.7	79.8	85.3

Table 3: CoT results for Embodied Emotion Detection. Llama: Llama-3.1-8B. DeepSeek: DeepSeek-R1-Distilled-Llama-8B. The *2-step*, *2-step-simple*, and *3-step* subscripts indicate the type of prompt accompanying the model in that run. Prompt details are in Table 6.

Chain-of-Thought (CoT) Analysis. Table 3 shows that CoT prompting enhanced performance to competitive levels with larger models in the experiments of Table 2, particularly benefiting distilled reasoning models like DeepSeek-R1-8B (distilled on Llama). The DeepSeek 8B model using simple 2-step prompts ($\text{DeepSeek}_{\text{2-step-simple}}$) achieved results within 6.7 F1-points of its larger 70B counterpart ($\text{DeepSeek}_{\text{simple}}$) and 2.7 F1-points of GPT 3.5. Deeper reasoning processes proved more effective, with 3-step CoT consistently outperforming 2-step variants across both models. Finally, simplified prompting substantially improved CoT performance, yielding F1-score increases of 6.7 ($\text{Llama}_{\text{2-step-simple}}$ vs. $\text{Llama}_{\text{2-step}}$) and 15.3 ($\text{DeepSeek}_{\text{2-step-simple}}$ vs. $\text{DeepSeek}_{\text{2-step}}$).

Error Analysis. To investigate model failures in the zero-shot experiments using the *simple* prompt setting with Llama-3.1-70B, we analyzed incorrect predictions. We found a pronounced false-positive bias, accounting for 93.3% of all errors. A manual inspection of 100 false-positive cases revealed

Model	F1	F1-J	F1-Sa	F1-F	F1-A	F1-D	F1-Su
Llama	31.6	39.4	43.6	26.6	32.2	19.1	28.5
DeepSeek	28.4	43.3	35.7	33.1	23.1	14.8	20.2
BWS_{4N}	41.8	62.3	57.7	37.9	28.1	30.1	33.9
BWS_{12N}	44.6	67.1	59.2	44.3	38.6	19.0	39.3
BWS_{36N}	**50.6**	66.7	**64.7**	48.0	**53.2**	22.0	**48.9**
BWS_{48N}	49.8	68.0	62.8	48.2	51.3	24.8	43.8
BWS_{72N}	49.5	**68.5**	64.5	46.2	51.6	20.5	45.6
BERT	49.6	68.2	57.5	**50.1**	30.2	**56.1**	35.7

Table 4: Results for Emotion Classification. Llama: Llama-3.1-8B. DeepSeek: DeepSeek-R1-Distilled-Llama-8B. BWS: Automatic BWS with Llama-3.1-8B. The first column F1 is the macro-averaged score, followed by F1-score F1-x, where J - *Joy*, Sa - *Sad*, F - *Fear*, A - *Anger*, D - *Disgust*, and Su - *Surprise*.

three main patterns. First, 17% of cases involved referenced body parts that were present in the experience or expression without acting, as in "*tears falling down the **face***." Second, 42% of errors stemmed from body parts performing functional or physiological roles within emotional contexts, such as eyes closing when "*blackness crept across his **eyes***," a natural physiological reaction associated with the character passing away within the context. Finally, 41% of errors involved metaphorical or idiomatic expressions. These included cases where emotional embodiment was implied but not explicitly stated ("*I couldn't believe my **eyes**,*" implying widened eyes in surprise but not explicitly describing this action), expressions symbolically referring to emotional states without literal physical embodiment ("*a straw that broke my **back***"), or purely locational expressions involving body parts without any action ("*thoughts racing through my **head***"). These nuanced distinctions highlight the model's challenges in accurately interpreting metaphorical, symbolic, and non-embodied references.

4.2 Embodied Emotion Classification

Our experimental results demonstrate notable performance disparities across prompting strategies and model architectures. We use the Llama-3.1-8B as the LLM interpreter for best-worst scaling (BWS). In Table 4, the first section shows the performance of zero-shot large language models, including Llama-3.1-8B and DeepSeek-R1-8B. The second section shows BWS results with different numbers of tuples. And the last section shows a fine-tuned BERT model for comparison (details in Appendix C). We see that BWS exhibits superior performance even with smaller tuple configurations ($4N$), exceeding Llama-3.1-8B by 10.2 points. Performance improves consistently as the number of tuples increases from $2N$ to $36N$ (40.2 to 50.6), suggesting enhanced classification from expanded pairwise comparisons. We hypothesize that this expansion helps the model better weigh emotional significance in text, improving classification accuracy. Notably, the best result comes from the expansion to $36N$ tuples, with the F1-score beating the supervised method BERT by 1 point. Our experimental results show that as we keep increasing N, the performance will reach a plateau as evidenced by $48N$ and $72N$ (see Appendix B).

Error Analysis. To better understand model limitations, we conducted a qualitative error analysis on the misclassified cases. We identified several consistent failure modes, including the model's difficulty in interpreting emotionally complex inputs or making reliable distinctions between closely related emotional states. In one case, the model predicted *Joy*, even though the embodied expression *"Ryan ducks his **head** down to his notebook"* signaled *Fear*. This misclassification likely resulted from the influence of nearby positive context, such as *"Brendon waves and smiles"*, which distracted the model from the emotion-relevant phrase. In another case, the vivid scene *"the age-old rock tradition of holding up lighters spread across the 28,000 person deep crowd . . . lighting up the entire audience . . . the hair on my **arms** started to raise"* was misclassified as *Joy*, despite strong physiological cues such as raised arm hair that more closely reflect *Surprise*. This highlights the model's tendency to prioritize surface-level celebratory language over conflicting embodied cues.

5 Conclusion

This work advances embodied emotion recognition through three main contributions. First, we created a new dataset called CHEER-Ekman by extending the CHEER dataset with Ekman emotion labels to better understand the connection between bodily expressions and emotional states. Second, we demonstrated that best-worst scaling outperforms both prompted LLMs and fine-tuned BERT, showing the potential for emotion classification without task-specific training. Finally, we found that simplified language and chain-of-thought reasoning significantly improve LLM performance in embodied emotion detection, enabling smaller models to achieve competitive results.

Limitations

While our approach demonstrates a promising advancement in embodied emotion detection using LLMs and the best-worst scaling technique, several limitations warrant consideration.

First, a key observation in our embodied emotion detection task was that simplifying prompts significantly improved model performance. While these findings may suggest enhanced efficiency through linguistic streamlining, they simultaneously introduce concerns about potential overfitting to these simplified phrasings. Simplified prompts may inadvertently prioritize more explicit expressions of embodied emotion over subtler or more figurative language, meaning the models might learn to recognize patterns specific to the prompt structure rather than generalizing to a wide variety of natural language expressions.

Second, the CHEER-Ekman dataset is relatively small, consisting of only 1,350 sentences. This limited size stems from our decision to annotate only the sentences already identified as containing embodied emotions in the original CHEER dataset. This selective annotation was intended to efficiently focus our efforts on instances most relevant to embodied emotion, but it may introduce a bias towards positive examples.

Finally, when addressing emotion classification via best-worst scaling, the scalability and computational overhead of this methodology present challenges: while higher tuple quantities lead to higher accuracy, they also impose significant computational costs. Due to time and computational constraints, we utilize a smaller model, which may lead to suboptimal results for higher-order tuples. Additionally, the limited context window prevents us from effectively implementing a few-shot setting, further impacting performance in scenarios requiring extended context understanding.

Acknowledgments

We thank the CincyNLP group for their constructive comments. We also thank the anonymous ACL reviewers for their valuable suggestions and feedback.

References

Christopher Bagdon, Prathamesh Karmalkar, Harsha Gurulingappa, and Roman Klinger. 2024. "you are an expert annotator": Automatic best–worst-scaling annotations for emotion intensity modeling. In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2024)*.

Alan Cowen and Dacher Keltner. 2017. Self-report captures 27 distinct categories of emotion bridged by continuous gradients. *Proceedings of the National Academy of Sciences of the United States of America*, 114.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, and Peiyi Wang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019)*.

Paul Ekman. 1992. Are there basic emotions? *Psychological Review*, 99(3).

Deepanway Ghosal, Navonil Majumder, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. 2020. COSMIC: COmmonSense knowledge for eMotion identification in conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2020 (Findings of EMNLP 2020)*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, and Alex Vaughan. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Svetlana Kiritchenko and Saif Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*.

Ronak Kosti, Jose M Alvarez, Adria Recasens, and Agata Lapedriza. 2019. Context based emotion recognition using emotic dataset. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(11).

George Lakoff and Mark Johnson. 1999. *Philosophy in the flesh: the embodied mind and its challenge of western thought*. Basic Books. OCLC: 302020239.

Gyeongeun Lee, Christina Wong, Meghan Guo, and Natalie Parde. 2024. Pouring your heart out: Investigating the role of figurative language in online expressions of empathy. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*.

Qintong Li, Piji Li, Zhaochun Ren, Pengjie Ren, and Zhumin Chen. 2022. Knowledge bridging for empathetic dialogue generation. In *Proceedings of the 36th AAAI conference on artificial intelligence (AAAI 2022)*.

Zhengyan Li, Yicheng Zou, Chong Zhang, Qi Zhang, and Zhongyu Wei. 2021. Learning implicit sentiment in aspect-based sentiment analysis with supervised contrastive pre-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*.

Zhiwei Liu, Kailai Yang, Qianqian Xie, Tianlin Zhang, and Sophia Ananiadou. 2024. Emollms: A series of emotional large language models and annotation tools for comprehensive affective analysis. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2024)*.

Saif Mohammad. 2018. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval 2018)*.

Paula M. Niedenthal. 2007. Embodying Emotion. *Science*, 316(5827):1002–1005.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*.

Sahand Sabour, Siyang Liu, Zheyuan Zhang, June Liu, Jinfeng Zhou, Alvionna Sunaryo, Tatia Lee, Rada Mihalcea, and Minlie Huang. 2024. EmoBench: Evaluating the emotional intelligence of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*.

Carlo Strapparava and Rada Mihalcea. 2007. SemEval-2007 task 14: Affective text. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007)*.

Weixiang Zhao, Zhuojun Li, Shilong Wang, Yang Wang, Yulin Hu, Yanyan Zhao, Chen Wei, and Bing Qin. 2024. Both matter: Enhancing the emotional intelligence of large language models without compromising the general intelligence. In *Findings of the Association for Computational Linguistics ACL 2024 (Findings of ACL 2024)*.

Yuan Zhuang, Tianyu Jiang, and Ellen Riloff. 2024. My heart skipped a beat! recognizing expressions of embodied emotion in natural language. In *Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2024)*.

A LLM Prompt for Embodied Emotion Detection & Embodied Emotion Classification

Table 5-7 presents the complete prompt templates employed in our experimental methodology. Table 5 details the prompts used for zero-shot embodied emotion detection experiments. The *Base* prompt closely replicates the methodology of Zhuang et al. (2024), with the sole modification being the use of "True" and "False" as decision tokens rather than "Yes" and "No". The *Simple* prompt maintains the fundamental logic while employing more straightforward language and structure to reduce cognitive complexity.

Table 6 presents our chain-of-thought (CoT) prompt variants. The *2-step* implementation incorporates the dual criteria from the Base prompt as explicit reasoning steps. The *3-step* variant augments this with an initial body movement identification phase, designed to establish concrete context and facilitate more comprehensive reasoning. The *2-step simple* variant examines the effectiveness of linguistic simplification within the CoT framework.

Table 7 outlines the large language model (LLM) prompt utilized for emotion classification experiments with Llama-3.1-8B and DeepSeek-R1-8B models.

Throughout our prompt templates, we employ the following placeholder semantics:

- "`<sentence|>`" denotes the target sentence containing the body part for evaluation.
- "`<bdypart|>`" indicates the specific body part instance within the sentence.
- "`<preceed|>`" represents up to three preceding context sentences, when available.

This placeholder convention remains consistent across all experimental tasks presented in this research.

B Best-Worst Scaling

Experiment Setup. Best-worst scaling (BWS) is a comparative annotation method where annotators select the best and worst items from a given set, typically a 4-tuple. This approach efficiently derives pairwise comparisons, as selecting the best and worst items provides information about most item relationships within the set. A single annotation with best-worst scaling is equivalent to an annotation with 6 pairwise comparisons. Hence, using BWS allows for fewer inferences with the same result. This is particularly beneficial for identifying emotion intensity or emotion classification.

These 4-tuples are assembled from the test data and then are presented to LLMs using the prompt in Table 8. The model then picks one instance that most represents and one instance that least represents some property (in our case, this would be one of the six Ekman's (1992) emotions).

Once multiple 4-tuples are annotated, a simple counting procedure generates numerical scores, allowing items to be ordered according to their relevance to the given property. The score is calculated using the formula $\frac{\#Best - \#Worst}{\#Overall}$, where the $\#Best$ and $\#Worst$ represent the number of times a sentence is ranked Best or Worst, respectively; and $\#Overall$ denotes the number of occurrences of the sentence across all 4-tuples. This approach captures a continuous measure, reflecting the relative intensity of a sentence within the given category.

With this method, the LLMs can perform accurate annotations. The results from the annotations will be calculated to get a BWS intensity score across 6 Ekman emotions. The emotion with the highest intensity score will be chosen as the predicted label. This classification process can be represented by the following expression:

$$\hat{e} = \arg\max_{e_i \in E} S(e_i)$$

where e_i corresponds to each emotion, E is the set of all emotions, and $S(e_i)$ is the intensity score w.r.t. to such emotion. The resulting predictions are compared with the labels from the CHEER-Ekman dataset to assess performance using several metrics, with a particular focus on the F1-score. The approach of increasing the number of tuples to enhance performance was proposed by Bagdon et al. (2024).

Along with the embodied emotion classification prompt, we also incorporate two additional placeholder semantics:

- "`<textid|>`" denotes the unique instance ID from the dataset. This id helps the model easily pick out its answer from the sentence tuple when inferencing.
- "`<emo|>`" denotes the specific emotion required for ranking.

This placeholder convention remains consistent across all experimental tasks presented in this research.

Performance Plateau. Figure 7 illustrates the relationship between tuple count and F1-score performance in our BWS experiments. The results demonstrate significant performance improvements up to $24N$ tuples, reaching optimal performance at $36N$. Beyond this threshold, performance degradation is observed, with decreased F1-scores at both $48N$ and $72N$ tuple configurations. This pattern suggests a clear upper bound for effective tuple scaling in BWS implementations.

C BERT Experiment Setup

For the embodied emotion classification task discussed in Section 3.3, in addition to BWS, we fine-tuned BERT as a reference benchmark (Devlin et al., 2019). Inputs were constructed by concatenating the preceding context, main text, and referenced body part. We set the maximum sequence length to 512 and use a batch size of 16. The model was trained with the AdamW optimizer, a learning rate of 2e-5, and evaluated over 15 training epochs. Cross-entropy loss was used as the objective, with tokenization performed using truncation and padding. All runs were conducted with 5 seed values starting from 41 to 45 for reproducibility, and final results were averaged across these five runs.

D Data and Results Analysis

Figure 3 - 6 present a more detailed analysis of the CHEER-Ekman dataset and evaluation of models' performance. Figure 3 illustrates the frequency of the top 10 body parts associated with each emotion, where the size of the bubble reflects the co-occurrence of the body part and emotion pair. Notably, the body parts *face*, *eye*, *head*, *hand*, and *throat* appear consistently across the top 10 body parts across all emotions, with the highest frequency observed in *face*, *eye*, and *head*.

Figure 4 illustrates the classification performance of the 10 most frequent body parts, with their frequency and corresponding accuracy across the three language models: Llama-3.1-8B, DeepSeek-R1-8B, and fine-tuned BERT. As expected, the fine-tuned BERT model consistently outperforms both Llama and DeepSeek for the most frequent body parts. Generally, the fine-tuned BERT model outperforms both zero-shot Llama and Deepseek, achieving an average accuracy increase of 11.7 over Llama and 14.8 over DeepSeek across the top 10 frequent body parts.

Figure 5 and 6 present confusion matrices comparing the models' predicted emotions against the ground truth emotion for Llama and DeepSeek, respectively. When comparing the two figures, a notable pattern emerges. In Figure 5, strong activations across the diagonal indicate Llama's attempt to predict emotions accurately without bias towards any one emotion. In Figure 6, however, we observe a prominent concentration in *Joy* in the DeepSeek model.

Task	Prompt Template
Base	Please determine if a body part is involved in any embodied emotion. Specifically, a body part is involved in some embodied emotion if both conditions below are satisfied: 1) The physical movement or physiological arousal involving the body part is evoked by emotion. 2) The physical movement, if there is any, has no other purpose than emotion expression. Answer "True" if the body part is involved in any embodied emotion, and "False" otherwise. Preceding Context: `<preceed|>` Sentence: `<sentence|>` Body part: `<bdypart|>` Answer:
Simple	Decide if a body part is used purely to express emotion. Ask: - Did emotion cause the body part's movement/response? - Was the movement ONLY for expressing emotion (no other reason)? If both are true, say "True." Else, say "False." Preceding Context: `<preceed|>` Sentence: `<sentence|>` Body part: `<bdypart|>` Answer:

Table 5: Zero-shot templates for different tasks.

Setting	Prompt Template
2-Step	Please determine if a body part is involved in any embodied emotion. First, answer Condition 1: Is the body part's movement/arousal caused by emotion? Then, answer Condition 2: Does the movement lack non-emotional purposes? If both of those conditions are true, answer "True." Otherwise, answer "False." Please reason step-by-step for your answer. Here is the question: Preceding Context: `<preceed|>` Sentence: `<sentence|>` Body part: `<bdypart|>`
3-Step	Please determine if a body part is involved in any embodied emotion. Specifically, a body part is involved in some embodied emotion if both conditions below are satisfied: Before answering, reasoning step-by-step 1. Identify the body part mentioned. 2. Check if emotion directly caused its movement/arousal. 3. Verify if the movement has no functional purpose. Only if all of the above are true, answer "True." Otherwise, answer "False." Here is the question: Preceding Context: `<preceed|>` Sentence: `<sentence|>` Body part: `<bdypart|>`
2-Step Simple	Decide if a body part is used purely to express emotion. Ask: - Did emotion cause the body part's movement/response? - Was the movement ONLY for expressing emotion (no other reason)? If both are true, say "True." Else, say "False." Before answering, give your reasoning step-by-step. Preceding Context: `<preceed|>` Sentence: `<sentence|>` Body part: `<bdypart|>`

Table 6: Chain of Thought (CoT) prompt templates for different settings.

Prompt Template
Classify the emotion expressed by the body part in a sentence into one of six categories: "Joy", "Sadness", "Anger", "Fear", "Surprise", or "Disgust". Preceding Context: <preceed|> Sentence: <sentence|> Body part: <bdypart|> Answer:

Table 7: Emotion classification prompt template.

Prompt Template
You are an expert annotator specializing in emotion recognition. Rank the following examples based on how much <emo|> the specified body part exudes in the text. Instructions: - Use only the Preceding Text for context. - Identify which example conveys the MOST <emo|> and which conveys the LEAST <emo|> based on the body part mentioned. - Do not repeat the text. Only provide the Example numbers in the specified format. Example: <textid|> Preceding Context: <preceed|> Sentence: <sentence|> Body part: <bdypart|> Example: <textid|> Preceding Context: <preceed|> Sentence: <sentence|> Body part: <bdypart|> Example: <textid|> Preceding Context: <preceed|> Sentence: <sentence|> Body part: <bdypart|> Example: <textid|> Preceding Context: <preceed|> Sentence: <sentence|> Body part: <bdypart|> Format your response as: Most <emo|> Example: Least <emo|> Example:

Table 8: BWS-Emotion classification prompt template.

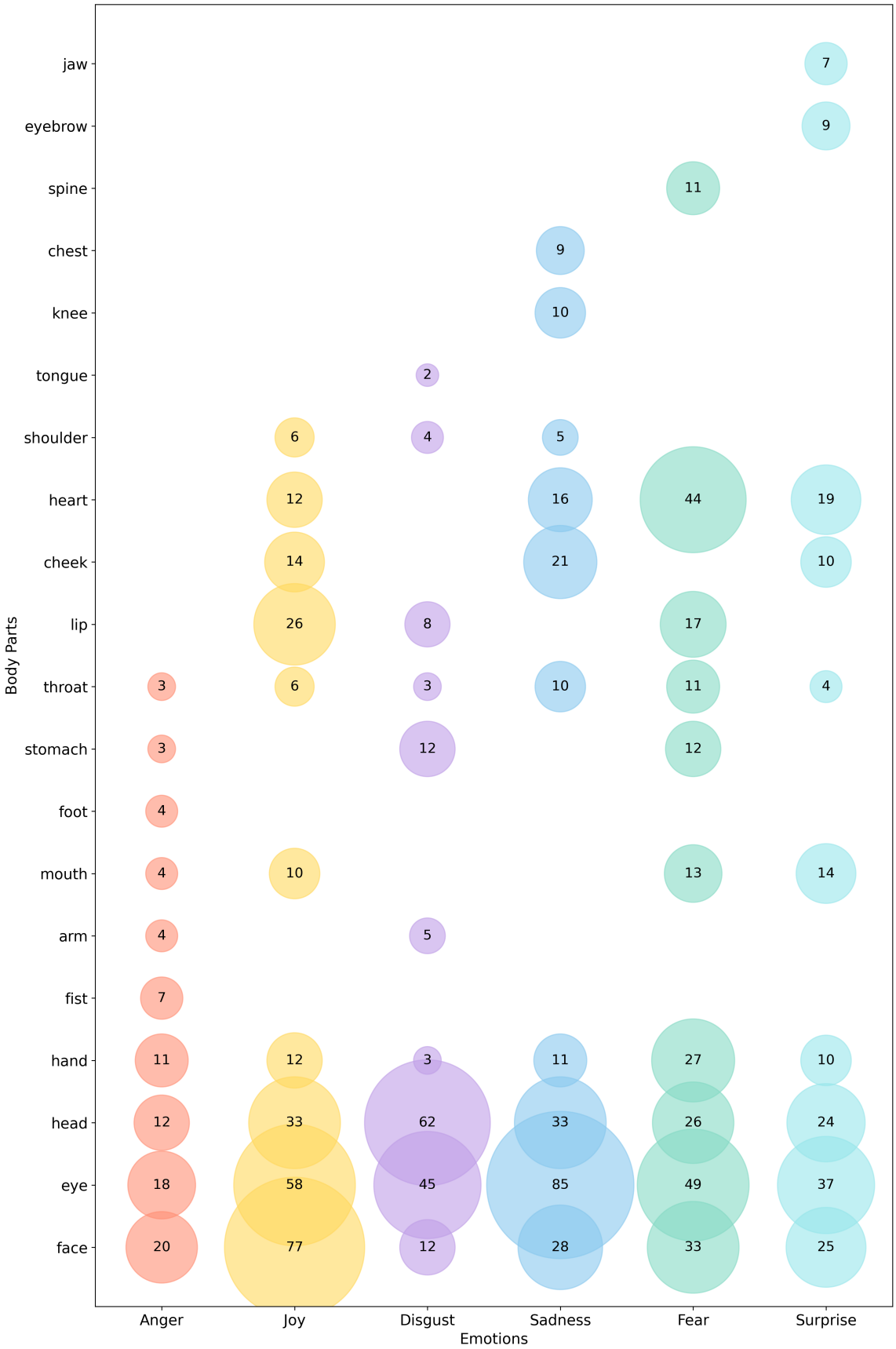


Figure 3: Frequency of top 10 body parts for each emotion.

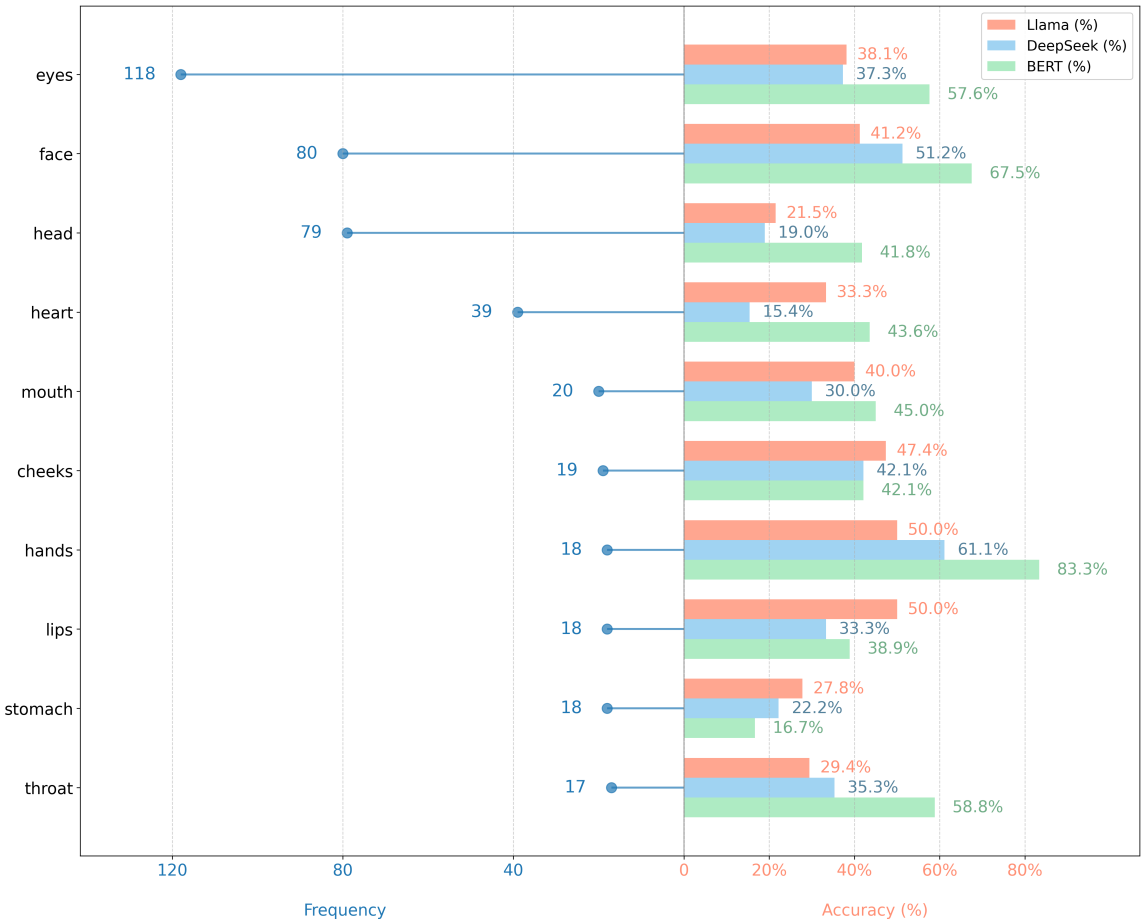


Figure 4: Embodied emotion classification of top 10 frequent body parts with frequency (left) and accuracy (right) comparing zeroshot with Llama, DeepSeek and finetuned BERT.

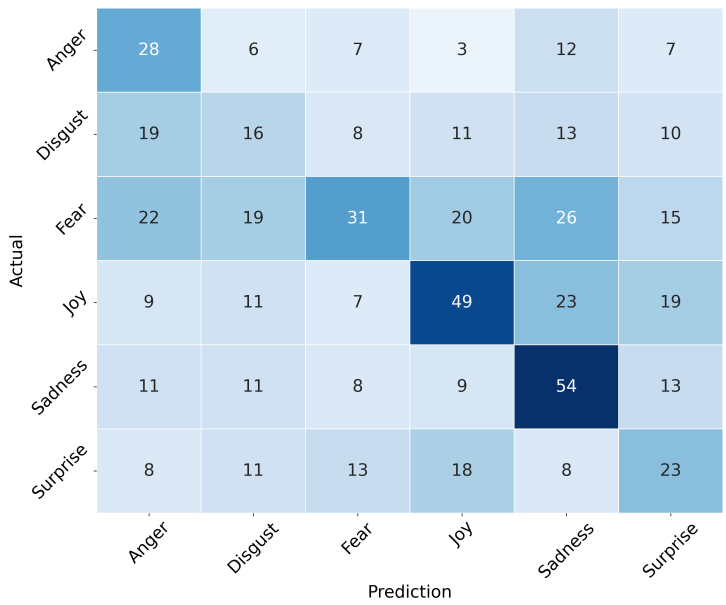


Figure 5: Confusion matrix of Llama predictions.

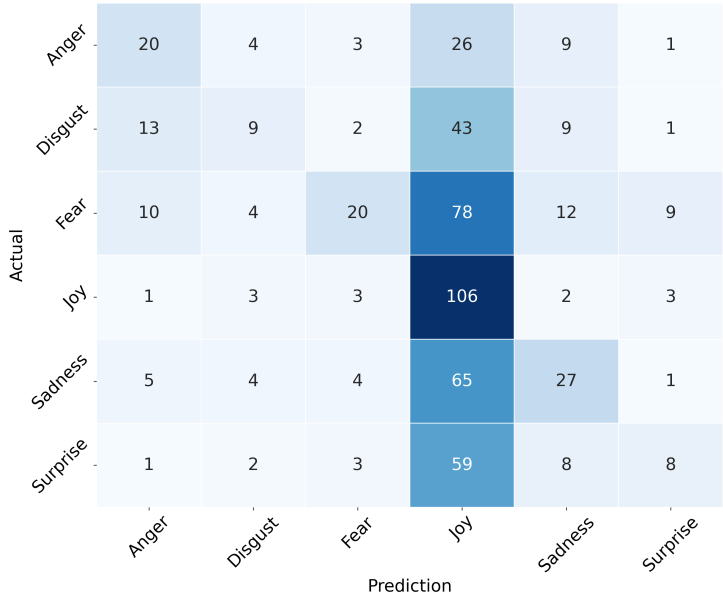


Figure 6: Confusion matrix of DeepSeek predictions.

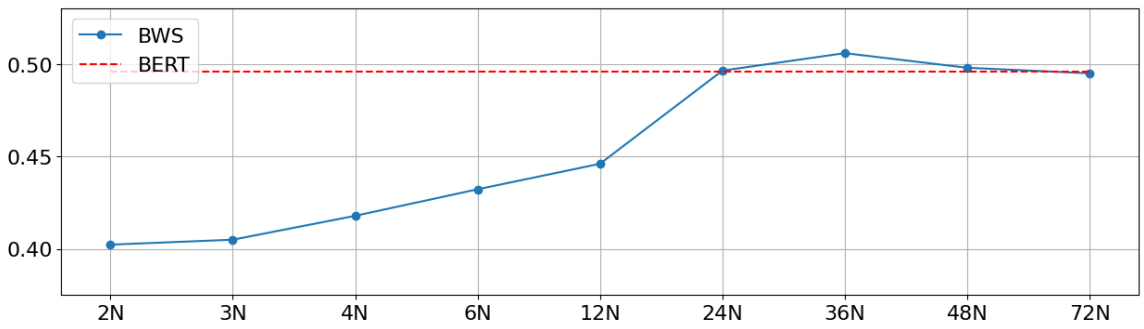


Figure 7: F1-score trends for BWS when increasing the number of tuples from 2N to 72N (where N is the total number of instances to be classified). BERT's performance is shown as a reference baseline.

ScanEZ: Integrating Cognitive Models with Self-Supervised Learning for Spatiotemporal Scanpath Prediction

Ekta Sood
University of Colorado Boulder
ekta.sood@colorado.edu

Prajit Dhar
University of Marburg
dhar@uni-marburg.de

Enrica Troiano
HK3Lab
enrica.troiano@hk3lab.ai

Rosy Southwell
University of Colorado Boulder
rosy.southwell@colorado.edu

Sidney K. D'Mello
University of Colorado Boulder
sidney.dmello@colorado.edu

Abstract

Accurately predicting human scanpaths during reading is vital for diverse fields and downstream tasks, from educational technologies to automatic question answering. To date, however, progress in this direction remains limited by scarce gaze data. We overcome the issue with ScanEZ, a self-supervised framework grounded in cognitive models of reading. ScanEZ jointly models the spatial and temporal dimensions of scanpaths by leveraging synthetic data and a 3-D gaze objective inspired by masked language modeling. With this framework, we provide evidence that two key factors in scanpath prediction during reading are: the use of masked modeling of both spatial and temporal patterns of eye movements, and cognitive model simulations as an inductive bias to kick-start training. Our approach achieves state-of-the-art results on established datasets (e.g., up to 31.4% negative log-likelihood improvement on CELER L1), and proves portable across different experimental conditions.

1 Introduction

Research at the intersection of cognitive science, computer vision (CV), and natural language processing (NLP) shows the potential for modeling human reading comprehension through eye-tracking data. Many downstream tasks, like entity recognition (Hollenstein and Zhang, 2019), summarization (Stiennon et al., 2020; Wu et al., 2021), and question answering (Sood et al., 2023), have particularly benefited from the use of scanpaths (i.e., records of eye movements on text), which provide signal about the readers' cognitive processes (Rayner, 1998) as well as the features of the text being read (Scozzaro et al., 2024; Wiechmann et al., 2022).

Scanpath datasets, however, are hard to obtain, and their scarcity represents a bottleneck for training gaze-aware architectures (Kümmerer et al., 2016). To solve this problem, past research has taken either of two directions: exploring self-supervised learning (SSL) techniques to predict eye gaze using limited labeled samples (Islam et al., 2021, i.a.), or augmenting existing samples with synthetic ones (Khurana et al., 2023). Works based on these approaches circumvented data shortages, but analyzed scanpath patterns to a limited extent: they modeled spatial dynamics (i.e., the order of words being fixated) that implicitly contain the chronology of fixations; yet, they overlooked *fixation duration* (i.e., how long the eyes dwell on a word). Duration is a key temporal aspect of reading because it links to word processing difficulty (Rayner, 1998; Clifton Jr et al., 2007; Vasishth et al., 2012). Nevertheless, related NLP research hardly incorporates this and other core linguistic and physiological factors that are well captured by cognitive models of reading behavior (Reichle et al., 2003; Engbert et al., 2005; Salvucci, 2001) – e.g., the effects of syntax complexity on word fixation (Cook and Wei, 2019), and the limits of visual range on eye movement (McDonald and Shillcock, 2003; McNamara and Magliano, 2009).

In this paper, we fill these gaps by combining data augmentation and SSL techniques: we propose ScanEZ, a SSL framework that uses synthetic data (grounded in cognitive models of reading) to predict gaze trajectories in terms of both spatial and temporal aspects, including fixation duration. With respect to data augmentation, ScanEZ uses the E-Z Reader cognitive model (Reichle et al., 2003) to derive synthetic gaze samples from existing corpora. On the modeling side, the framework jointly learns the spatial (x, y) and temporal (t) coordinates of gaze by employing a masked gaze modeling objective. Masked language modeling, a well-established approach in NLP and CV (Devlin et al., 2019; Zhang et al., 2022; Kwon et al., 2022), is effective for capturing rich sequence dependencies. We demonstrate its utility to predict full 3-D spatiotemporal trajectories of scanpaths, which inherently rely on textual dependencies. Ex-

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1132–1142
July 27 - August 1, 2025

perimental results demonstrate strong performance across datasets, including CELER L1 (Berzak et al., 2022), where ScanEZ surpasses the state of the art (Deng et al., 2023) by up to 31.4% (negative log-likelihood), and ZuCo 1.0 (Hollenstein, 2018), with a 58.6% improvement. We also evaluate our approach with fixation duration and fixation location accuracy, two eye-movement metrics which assess the model's ability to capture *how long* the eyes dwell on a word, reflecting temporal attention in reading, and *where* the eyes fixate, reflecting spatial attention.

In sum, this work advances scanpath prediction with the following contributions: (1) a novel framework to model spatial and temporal dimensions simultaneously, integrating cognitive data with scalable NLP techniques (notably, masked language modeling); (2) state-of-the-art scanpath prediction results, complemented by analyses to assess cross-domain generalization performance, and the role of synthetic data pre-training and human-data fine-tuning; (3) additional model evaluation based on eye-tracking metrics that capture the interaction between text and readers.

2 Related Work

Reading comprehension involves multiple cognitive processes (Kintsch, 1998), from word and sentence processing to integrating the text with a reader's prior knowledge (Kintsch, 1994; McNamara and Magliano, 2009). These processes unfold chronologically, but they extend beyond simple left-to-right reading, as eye movements have non-linear patterns of word fixation and skipping (Engbert et al., 2004). Accurately modeling both spatial and temporal aspects of eye movements is thus essential to understand reading behavior.

Empirical evidence is plentiful for the link between specific eye movement features (fixation duration, location, etc.) and key linguistic and cognitive phenomena, for which specific measures have been proposed (Cook and Wei, 2019; McDonald and Shillcock, 2003; Shain, 2024; Just and Carpenter, 1976; Rayner et al., 1998). These features are also captured by computational cognitive models which render explicit the relationship between cognitive processes, like word recognition, and physical actions, such as word fixations. Among these, E-Z Reader (Reichle et al., 2003) simulates the control of eye movements on a given text. E-Z Reader has been shown to approximate human gaze behavior across a range of datasets and experimental conditions (Mancheva et al., 2015; Reichle and Drieghe, 2013; Reichle and Sheridan, 2015, i.a.,). Its predictive validity has been further supported by evaluations against human eye-tracking data, including comparisons on the CELER dataset (Deng et al., 2023). Moreover, E-Z Reader-based simulations over the CNN and DM corpora have demonstrated utility in downstream tasks, achieving top performance in text saliency prediction when modeled during pre-training (Sood et al., 2020).

While replicating core reading patterns, E-Z Reader relies on handcrafted features that limit its scalability for AI applications. Deep learning offers alternatives which can closely approximate human reading patterns. For instance, SCANDL uses a sequence-to-sequence diffusion model to capture gaze-text interactions (Bolliger et al., 2023), SP-EyeGAN simulates raw eye-tracking data with generative adversarial networks (Prasse et al., 2023), and Eyettention (Deng et al., 2023) introduces a dual-sequence encoder-decoder architecture with cross-attention mechanisms to align linguistic and temporal sequences to predict the next fixation location. SCANDL and SP-EyeGAN further address the paucity of eye-tracking resources with their ability to generate synthetic data. In part, the lack of data is also mitigated by SSL (Ericsson et al., 2022) approaches, which derive gaze features from unlabeled samples. Contrastive and predictive objectives, for instance, have been used to capture statistical regularities in fixation patterns to support downstream tasks (Prasse et al., 2024). However, SSL still requires large-scale data.

In this paper, we adopt SSL like SCANDL and SP-EyeGAN, while pre-training on fixations generated with the E-Z Reader cognitive model. This way, we push masked language modeling for reading gaze representation to be data-rich and cognitively-driven (Sood et al., 2020). Notably, we extend the approach of Deng et al. (2023), who model spatial scanpath coordinates for scanpath prediction, by simultaneously incorporating temporal coordinates (fixation durations). We directly compare our results to theirs, as Eyettention represents the current state of the art. To enrich model evaluation, we further employ experimental eye-movement metrics. As part of this framework, we use synthetic corpora introduced in prior work, relying on the established validity of the E-Z Reader model as a synthetic data generator.

3 ScanEZ

We propose a framework that adapts a BERT-style transformer for trajectory modeling, using a masked sequence prediction objective and synthetic data pre-training to enable representation learning without large labeled datasets.

Input Representation and Masking. Inputs to the model are gaze data parsed into sequences of fixations, each represented by three features describing the trajectory of gaze over text: the x- and y-coordinates (space) and fixation duration t (time). For preprocessing, each input sequence $X \in \mathbb{R}^{T \times F}$, where T denotes the number of time steps and $F = 3$ the number of features, is normalized to zero mean and unit variance. Normalization ensures numerical stability during training and harmonizes the distributions of real and synthetic data.

During pre-training, a subset of input features is masked: a fixed proportion of the sequence is obscured, generating X_{masked}, where partial information is visible to the model. In the prediction task, the model predicts the masked values (x, y, t) based on surrounding context, learning dependencies across the spatial and temporal dimensions.

Model Architecture. The network consists of an embedding layer which projects the input features (x, y, t) into a latent space of dimension d, creating dense embeddings that preserve spatiotemporal relationships. Sinusoidal positional encodings are added to the embeddings, ensuring that the model distinguishes between otherwise position-invariant elements in the sequence. The model also comprises L transformer layers with a multi-head self-attention mechanism (h attention heads attend to relationships across time steps and between spatial and temporal features), and a fully connected feedforward network with two linear layers and ReLU activations (which capture interactions between features).

Training Objective. The pre-training task is to reconstruct masked portions of the input sequence. Given a masked input X_{masked}, the model predicts the values of the masked features (x, y, t):

$$\mathcal{L} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} (X_i - \hat{X}_i)^2 \quad (1)$$

where $\mathcal{M}$ denotes the set of masked indices, X_i the ground truth, and $\hat{X}_i$ the predicted values. This loss encourages the model to learn embeddings that capture both local and global spatiotemporal dependencies in sequential gaze data.

We pre-train the model on synthetic 3-D trajectories, and fine-tune its weights on real eye-tracking datasets – both sources of data are discussed next.

4 Experimental Setup

We now describe the datasets and evaluation protocols used for scanpath prediction. To better reflect real-world reading scenarios, we selected diverse benchmark datasets and adopted multiple evaluation settings, including three distinct data-splitting approaches. Further details on the data and evaluation metrics are in appendix A.1.

Data. As illustrated in Figure 1, we pre-train ScanEZ on E-Z Reader simulations, which were generated by Sood et al. (2020) from the CNN and Daily Mail corpus (Hermann et al., 2015). For fine-tuning, we use two human sentence-reading datasets with the same train/validation splits for the scanpath prediction task as in Deng et al. (2023). These datasets vary in complexity, domain, and participant diversity, enabling in-domain and cross-domain robustness tests: *CELER L1* (Berzak et al., 2022), with native (L1) English speakers reading single-line sentences from Wall Street Journal (Marcus et al., 1993): and *ZuCo 1.0* (Hollenstein et al., 2018), with L1 English speakers reading sentences from the Stanford Sentiment Treebank (Socher et al., 2013) and Wikipedia relation extraction corpus (Culotta et al., 2006), with sentences displayed as multiple (maximum 7) lines. In addition, we use *EML* (Caruso et al., 2022), a dataset of fluent English speakers reading 5 complex educational texts, each of ≈1000 words, displayed as 50 full pages with ≈100 words on each, from which we extract 239 sentences.

Evaluation. We analyze performance in the four cross-validation task settings reported in Figure 1. *Participant split* (Part): Train/validate on distinct sets of participants. *Text split* (Text): Train/validate on distinct texts (i.e. sentences or pages)[1]. *Participant + Text split* (P.T.): No overlap in participants nor texts between training and validation. This is the most challenging condition for the within dataset evaluation settings (Deng et al., 2023). *Cross-dataset*: Train on a dataset and test on another, to assess cross-domain generalization. We keep the same 5-fold evaluation protocol across settings for robust performance estimates.

[1]We use *text* for any stimulus shown to participants. Sentence count per stimulus varies by dataset: Celer ≈ 1; ZuCo ≈ 2–3; EML ≈ 5.

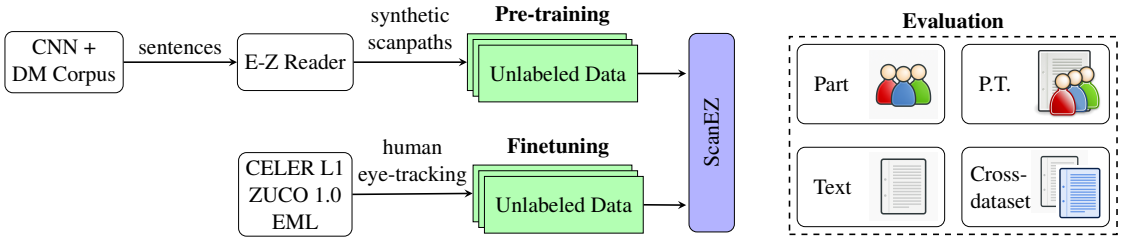


Figure 1: Overview of the workflow combining synthetic and human eye-tracking data for scan-path prediction. Synthetic scanpaths generated with the E-Z Reader model from CNN + DM texts are used in the pre-training phase of SCANEZ. The model is then fine-tuned on real human data (CELER, ZuCo 1.0, EML). We evaluate ScanEZ's performance in four cross-validation settings – Part(), Text(), P.T.(), and Cross-dataset() – described Section 4.

For spatial predictions, we report Negative Log-Likelihood (NLL), to measure how well the model fits observed scan paths, and Normalized Levenshtein Distance (NLD), quantifying alignment quality between predicted and actual sequences. For temporal predictions, we use NLL_t, as NLD considers the distance between spatial sequences (e.g., strings), which does not apply to the temporal dimension. To better observe eye movement characteristics, we also include fixation location accuracy (FLA) and fixation duration accuracy (FDA), which range from 0 (worst possible accuracy) to 1 (best).

5 Does ScanEZ Capture Where Readers Fixate, and When They Do So?

We answer this question by replicating the experiments of Deng et al. (2023) and observing ScanEZ performance on naturalistic (EML) data.[2]

Comparison with Eyettention. ScanEZ achieves better (i.e., lower) NLL and NLD under all experimental settings. For instance, on the Text split of CELER L1, it has a NLL of 1.603 compared to the 2.277 of Eyettention, and a NLD of 0.43, which is 0.142 points more than Eyettention; on the Part split, it brings an improvement of 0.712 points in NLL (NLL = 1.555, against Eyettention's 2.267) and of 0.149 points in NLD (NLD = 0.424 for ScanEZ, 0.573 for Eyettention). Even on P.T., which is our most challenging condition, ScanEZ surpasses Eyettention on CELER L1. Table 1 focuses on this setting, and shows that ScanEZ performs better by 0.77 and 0.147 points, with a NLL score of 1.524 and a NLD score of 4.421.

This improvement remains valid in the cross-dataset setup: on the P.T. split, our framework achieves NLL = 0.548 when trained on CELER L1 and tested on ZuCo 1.0, whereas Eyettention has a score of 2.613. On average, the P.T. NLL is reduced by 58.6% compared to the baseline when testing on ZuCo 1.0 across train-testing combinations.[3] These scores corroborate the value of masked-language modeling in scanpath prediction, promoting out-of-domain generalization.

Table 1 also provides evidence for the benefit of pre-training and fine-tuning ScanEZ: removing human-based fine-tuning brings NLL up to 3.035, which underperforms both ScanEZ and Eyettention, and models without pre-training are better than Eyettention (potentially thanks to the fine-tuning step) but still worse than ScanEZ (1.77 NLL). Similar results hold for the other metrics as well: ScanEZ outperforms its ablation alternatives with respect to FDA and FLA; moreover, it sets benchmark performance for the temporal prediction in terms of NLL_t.[4] Overall, fine-tuning enables the model to efficiently capture eye movement patterns in real eye-tracking data, and kick-starting training with E-Z Reader simulations supports robust initial representations. Put together, these components allow ScanEZ to deliver superior performance in spatial scanpath prediction, and efficiency in the modeling of its temporal dimension.

Evaluations on EML We analyze the performance of ScanEZ and its ablated variants on the EML dataset (Caruso et al., 2022). Once more, pre-training yields pronounced improvements across metrics (Table 1, bottom). An exception is FLA,

[2]We report direct comparisons only with Deng et al. (2023), as their work includes extensive evaluations against a broad range of models. This allows to interpret our findings in the light of those prior comparisons.

[3]A full by-setting breakdown of all results discussed in this section is in appendix A.2 (Table 3), including performance on train-test set combinations (Table 4), and ablation analyses to isolate the role of pre-training and fine-tuning (Table 5).

[4]These three metrics are unavailable for Eyettention.

Data	Model	NLL↓	NLD↓	NLL$_t$↓	FDA↑	FLA↑
CLR	Eyettention	2.297 ± 0.011	0.568 ± 0.004	—	—	—
	w/o Fine-tuning	3.035 ± 0.204	0.950 ± 0.009	1.403 ± 0.012	0.603 ± 0.004	0.284 ± 0.018
	w/o Pre-training	1.772 ± 0.304	0.547 ± 0.085	1.431 ± 0.150	0.611 ± 0.054	0.381 ± 0.054
	ScanEZ	**1.524 ± 0.042**	**0.421 ± 0.022**	**1.244 ± 0.027**	**0.646 ± 0.004**	**0.413 ± 0.009**
EML	w/o Fine-tuning	1.837± 0.025	0.761± 0.005	1.181± 0.011	0.797± 0.003	**0.589± 0.006**
	w/o Pre-training	1.217± 0.212	0.777± 0.004	1.177± 0.066	0.785± 0.012	0.491± 0.023
	ScanEZ	**0.996± 0.020**	**0.616± 0.013**	**1.083± 0.033**	**0.804± 0.005**	0.534± 0.012

Table 1: Top: comparison between our framework and Eyettention on the CELER L1 dataset (CLR) to our model trained on: only EZ-Reader data (w/o Fine-tuning), only human data (w/o Pre-training), and with both pre-training on EZ-Reader and then fune-tuning on human data (ScanEZ). Bottom: evaluation using the EML dataset. ↓: the lower the score, the better. ↑: vice versa. All results refer to the P.T. setting.

significantly higher for the w/o Fine-tuning setting than for ScanEZ (p-value = 0.001, with an ANOVA test). This result is inconsistent with that on CELER L1: we attribute it to EML's more challenging data, which comprises multi-sentences texts rather than single sentence texts, making the task more complex.[5] While future work could investigate this insight, the results presented here maintain that ScanEZ effectively generalizes to various reading conditions.

6 Conclusion

We introduced ScanEZ, a self-supervised framework that addresses the challenge of scan path prediction under data scarcity conditions. Aimed at an explicit spatiotemporal modeling of eye movements, ScanEZ demonstrated robust performance and generalization abilities across datasets, experimental settings, and metrics, surpassing the state of the art on spatial predictions and benchmarking the prediction of fixation duration. Importantly, our results point to the benefits of the masked-language modeling approach we implemented, which opens up new research directions on eye-tracking data. As a matter of fact, since masked language modeling has not been as widely adopted for scanpath prediction as they have in NLP tasks, our finding that a BERT-based model yields superior performance constitutes a core contribution of this work. Our ablation analyses further showed the importance of real-data fine-tuning for a model pre-trained on synthetic gaze trajectories. Both components, which underpin the value of bridging cognitive model simulations with data-driven methodologies for gaze representation, allowed ScanEZ to effectively capture spatiotemporal dependencies.

[5]These FLA differences are analyzed in appendix A.2.

7 Limitations

We evaluated ScanEZ on two established open-source datasets (CELER L1 and Zuco 1.0). This experimental decision ensured comparability with previous work, specifically with Eyettention. However, to fully understand the generalizability of the model we proposed, we would need to conduct additional evaluations, in particular on datasets representing real-world reading scenarios – unlike CELER L1 and ZuCo 1.0, which were built in a lab environment. We took a step in this direction by reporting results on the EML data (drawn from a naturalistic e-learning setting). Yet, EML is not open source, which might complicate the reproduction of our results for other researchers.

Focusing on such results, we reported ScanEZ's success in modeling both spatial and temporal aspects of gaze. The comparison with past work, however, was only partial, as NLLt, fixation duration accuracy (FDA), and fixation location accuracy (FLA) are not reported for Eyettention.

Lastly, by addressing scanpath prediction, our work contributes to an established line of research that studies eye movements (e.g., Boccignone et al., 2019; de Belen et al., 2022; Mishra et al., 2018). As such, it represents a step towards evaluating gaze-informed models in applications related to reading behavior. To date, however, we still need to test ScanEZ on downstream tasks, to better understand the breadth of its potential.

8 Ethical Considerations

This work did not go through an ethical committee; still, we ensured compliance with ethical principles in data usage and methodology. On the one hand, we used artificial gaze trajectories – no real individuals can be reconstructed from it. On the other,

we relied on publicly available datasets and EML (all already peer-reviewed) where individuals are thoroughly anonymized.

Acknowledgements

This research was supported by the National Science Foundation (DRL 1920510 and 2019805). The opinions expressed are those of the authors and do not represent views of the funding agency.

References

Yevgeni Berzak, Chie Nakamura, Amelia Smith, Emily Weng, Boris Katz, Suzanne Flynn, and Roger Levy. 2022. CELER: A 365-Participant Corpus of Eye Movements in L1 and L2 English Reading. *Open Mind*, 6:41–50.

Giuseppe Boccignone, Vittorio Cuculo, and Alessandro D'Amelio. 2019. How to look next? a data-driven approach for scanpath prediction. In *International Symposium on Formal Methods*, pages 131–145. Springer.

Lena Bolliger, David Reich, Patrick Haller, Deborah Jakobi, Paul Prasse, and Lena Jäger. 2023. ScanDL: A diffusion model for generating synthetic scanpaths on texts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15513–15538, Singapore. Association for Computational Linguistics.

Megan Caruso, Candace Peacock, Rosy Southwell, Guojing Zhou, and Sidney D'Mello. 2022. Going Deep and Far: Gaze-based Models Predict Multiple Depths of Comprehension During and One Week Following Reading.

Charles Clifton Jr, Adrian Staub, and Keith Rayner. 2007. Eye movements in reading words and sentences. *Eye movements*, pages 341–371.

Anne E. Cook and Wei Wei. 2019. What Can Eye Movements Tell Us about Higher Level Comprehension? *Vision*, 3(3):45. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

Aron Culotta, Andrew McCallum, and Jonathan Betz. 2006. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 296–303, New York City, USA. Association for Computational Linguistics.

Ryan Anthony Jalova de Belen, Tomasz Bednarz, and Arcot Sowmya. 2022. Scanpathnet: A recurrent mixture density network for scanpath prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5010–5020.

Shuwen Deng, David R. Reich, Paul Prasse, Patrick Haller, Tobias Scheffer, and Lena A. Jäger. 2023. Eyettention: An Attention-based Dual-Sequence Model for Predicting Human Scanpaths during Reading. ArXiv:2304.10784 [cs].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Ralf Engbert, Reinhold Kliegl, and André Longtin. 2004. Complexity of eye movements in reading. *International Journal of Bifurcation and Chaos*, 14(02):493–503.

Ralf Engbert, Antje Nuthmann, Eike M. Richter, and Reinhold Kliegl. 2005. SWIFT: A dynamical model of saccade generation during reading. *Psychological Review*, 112(4):777–813.

Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M. Hospedales. 2022. Self-Supervised Representation Learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39(3):42–62.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Nora Hollenstein. 2018. Zuco, a simultaneous eeg and eye-tracking resource for natural sentence reading. *Scientific data*.

Nora Hollenstein, Jonathan Rotsztejn, Marius Troendle, Andreas Pedroni, Ce Zhang, and Nicolas Langer. 2018. ZuCo, a simultaneous EEG and eye-tracking resource for natural sentence reading. *Scientific Data*, 5(1):180291. Number: 1 Publisher: Nature Publishing Group.

Nora Hollenstein and Ce Zhang. 2019. Entity recognition at first sight: Improving NER with eye movement information. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1–10, Minneapolis, Minnesota. Association for Computational Linguistics.

Md. Rabiul Islam, Shuji Sakamoto, Yoshihiro Yamada, Andrew W. Vargo, Motoi Iwata, Masakazu Iwamura, and Koichi Kise. 2021. Self-supervised learning for reading activity classification. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5(3).

Marcel Adam Just and Patricia A. Carpenter. 1976. Eye fixations and cognitive processes. *Cognitive Psychology*, 8(4):441–480.

Varun Khurana, Yaman Kumar, Nora Hollenstein, Rajesh Kumar, and Balaji Krishnamurthy. 2023. Synthesizing human gaze feedback for improved NLP performance. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1895–1908, Dubrovnik, Croatia. Association for Computational Linguistics.

Walter Kintsch. 1994. Text comprehension, memory, and learning. *American Psychologist*, 49(4):294–303. Num Pages: 294-303 Place: Washington, US Publisher: American Psychological Association (US).

Walter Kintsch. 1998. *Comprehension: A Paradigm for Cognition*. Cambridge University Press.

Matthias Kümmerer, Thomas S. A. Wallis, and Matthias Bethge. 2016. Deepgaze II: reading fixations from deep features trained on object recognition. *CoRR*, abs/1610.01563.

Gukyeong Kwon, Zhaowei Cai, Avinash Ravichandran, Erhan Bas, Rahul Bhotika, and Stefano Soatto. 2022. Masked vision and language modeling for multi-modal representation learning. *arXiv preprint arXiv:2208.02131*.

Lyuba Mancheva, Erik D. Reichle, Benoît Lemaire, Sylviane Valdois, Jean Ecalle, and Anne Guérin-Dugué. 2015. An analysis of reading skill development using e-z reader. *Journal of Cognitive Psychology (Hove)*, 27(5):357–373. Epub 2015 Apr 9. PMID: 27148437, PMC4852752.

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

Scott A. McDonald and Richard C. Shillcock. 2003. Eye Movements Reveal the On-Line Computation of Lexical Probabilities During Reading. *Psychological Science*, 14(6):648–652. Publisher: SAGE Publications Inc.

Danielle S. McNamara and Joe Magliano. 2009. Chapter 9 Toward a Comprehensive Model of Comprehension. In *Psychology of Learning and Motivation*, volume 51 of *The Psychology of Learning and Motivation*, pages 297–384. Academic Press.

Abhijit Mishra, Pushpak Bhattacharyya, Abhijit Mishra, and Pushpak Bhattacharyya. 2018. Scanpath complexity: modeling reading/annotation effort using gaze information. *Cognitively Inspired Natural Language Processing: An Investigation Based on Eye-tracking*, pages 77–98.

Paul Prasse, David R. Reich, Silvia Makowski, Tobias Scheffer, and Lena A. Jäger. 2024. Improving cognitive-state analysis from eye gaze with synthetic eye-movement data. *Computers & Graphics*, 119:103901.

Paul Prasse, David Robert Reich, Silvia Makowski, Seoyoung Ahn, Tobias Scheffer, and Lena A. Jäger. 2023. Sp-eyegan: Generating synthetic eye movement data with generative adversarial networks. In *Proceedings of the 2023 Symposium on Eye Tracking Research and Applications*, ETRA '23, New York, NY, USA. Association for Computing Machinery.

Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372.

Keith Rayner, Erik D. Reichle, and Alexander Pollatsek. 1998. Eye Movement Control in Reading: An Overview and Model. *Eye guidance in reading and scene perception*, pages 243–268.

Erik D. Reichle and Denis Drieghe. 2013. Using EZ reader to examine word skipping during reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 39(4):1311.

Erik D. Reichle, Keith Rayner, and Alexander Pollatsek. 2003. The E-Z reader model of eye-movement control in reading: Comparisons to other models. *The Behavioral and Brain Sciences*, 26(4):445–476; discussion 477–526.

Erik D. Reichle and Heather Sheridan. 2015. EZ reader: An overview of the model and two recent applications. *The Oxford handbook of reading*, page 277.

Dario D. Salvucci. 2001. An integrated model of eye movements and visual encoding. *Cognitive Systems Research*, 1(4):201–220.

Calogero J. Scozzaro, Davide Colla, Matteo Delsanto, Antonio Mastropaolo, Enrico Mensa, Luisa Revelli, and Daniele P. Radicioni. 2024. Legal text reader profiling: Evidences from eye tracking and surprisal based analysis. In *Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context @ LREC-COLING 2024*, pages 114–124, Torino, Italia. ELRA and ICCL.

Cory Shain. 2024. Word Frequency and Predictability Dissociate in Naturalistic Reading. *Open Mind*, 8:177–201.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Ekta Sood, Fabian Kögel, Philipp Müller, Dominike Thomas, Mihai Bâce, and Andreas Bulling. 2023. Multimodal Integration of Human-Like Attention in Visual Question Answering. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2648–2658, Vancouver, BC, Canada. IEEE.

Ekta Sood, Simon Tannert, Philipp Mueller, and Andreas Bulling. 2020. Improving Natural Language Processing Tasks with Human Gaze-Guided Neural Attention.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Shravan Vasishth, Titus von der Malsburg, and Felix Engelmann. 2012. What can eye-tracking tell us about sentence processing. *WIREs Cognitive Science*.

Daniel Wiechmann, Yu Qiao, Elma Kerz, and Justus Mattern. 2022. Measuring the impact of (psycho-)linguistic and readability features and their spill over effects on the prediction of eye movement patterns. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5276–5290, Dublin, Ireland. Association for Computational Linguistics.

Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*.

Chaoning Zhang, Chenshuang Zhang, Junha Song, John Seon Keun Yi, Kang Zhang, and In So Kweon. 2022. A survey on masked autoencoder for self-supervised learning in vision and beyond. *arXiv preprint arXiv:2208.00173*.

A Appendix

A.1 Details on Experimental Setup

Data. Table 2 reports statistics on the data used to pre-train and fine-tune ScanEZ.

Pre-training	Sentences	Simulations
E-Z Reader: CNN	3.1M	10
E-Z Reader: Daily Mail	7.6M	10
Fine-tuning	Sentences	Participants
CELER L1	5457	69
ZuCo 1.0	700	12
EML	239	147

Table 2: Descriptive statistics of the used datasets. Top: synthetic datasets used for pre-training; bottom: human datasets used in our experiments.

Metrics. The two metrics FDA (Fixation Duration Accuracy) and FLA (Fixation Location Accuracy) are computed as follows:

$$\text{FDA} = 1 - \frac{|T_{\text{pred}} - T_{\text{ground}}|}{\max(|T_{\text{pred}} - T_{\text{ground}}|)}$$

where T_{pred} is the predicted fixation duration, and T_{ground} denotes the ground-truth fixation duration;

$$\text{FLA} = 1 - \frac{\sqrt{\begin{array}{c}(X_{\text{pred}} - X_{\text{ground}})^2 \\ + (Y_{\text{pred}} - Y_{\text{ground}})^2\end{array}}}{\max\left(\sqrt{\begin{array}{c}(X_{\text{pred}} - X_{\text{ground}})^2 \\ + (Y_{\text{pred}} - Y_{\text{ground}})^2\end{array}}\right)}$$

where X_{pred} and Y_{pred} respectively indicate the predicted fixation of x and the predicted fixation of y, and X_{ground} and Y_{ground} indicate the ground-truth fixation of x and the ground-truth fixation of y. Whenever the model perfectly predicts a scanpath, both FDA and FLA are set to 1, hence avoiding dividing by zero.

A.2 Results

We report the full results obtained by replicating Deng et al.'s experiments with both their Eyettention and our ScanEZ; in addition, we detail our ablation analyses, as well as our experiments on EML, a dataset that is not used in Deng et al. (2023): scanpath prediction performance on CELER L1 is in Table 3, broken down by the three settings of Text, Part and P.T.; Table 4 shows the cross-dataset results, providing evidence for ScanEZ's superior generalization abilities compared to Eyettention; Table 5 focuses on the contribution of the pre-training and fine-tuning steps of ScanEZ, each removed in the Doc, Part and P.T. settings; Table 6 does the same but on EML.

In Section 5, we noted that the advantage of fine-tuning ScanEZ on CELER L1 reflects in all metrics. On EML, however, FLA increases without fine-tuning. We thus conducted a small-scale analysis to study this difference between datasets, hypothesizing that scanpath length (i.e., the number of fixations) has an effect on FLA, as a space-related metrics. With a regression study, we found that this is indeed the case. Taking scanpath length as the independent variable and FLA as the dependent variable, we observed a significant effect across all datasets, including ZuCo 1.0: the positive t-scores in Table 7 denote that that longer scanpaths result

in higher FLA values. Notably, the t-score is higher on CELER L1 (92.36) than on EML (73.03), which indicates that longer scanpath sequences were easier to predict on the former dataset. This insight confirms that the more challenging data of EML make FLA improvements harder. Interestingly, the same conclusion can be drawn for FDA, although the increase of this metric in the "w/o Fine-tuning" setup was stable across datasets (cf. Table 1). Further analyses are therefore needed to better compare the two metrics, particularly regarding their relationship to scanpath features other than length.

Task	Model	NLL↓	NLD↓	NLL_t↓
Text	Eyettention	2.277 ± 0.005	0.572 ± 0.002	—
	ScanEZ	**1.603 ± 0.017**	**0.430 ± 0.047**	**1.269 ± 0.068**
Part	Eyettention	2.267 ± 0.005	0.573 ± 0.002	—
	ScanEZ	**1.555 ± 0.013**	**0.424 ± 0.025**	**1.235 ± 0.072**
P.T.	Eyettention	2.297 ± 0.011	0.568 ± 0.004	—
	ScanEZ	**1.524 ± 0.042**	**0.421 ± 0.022**	**1.244 ± 0.027**

Table 3: Performance on CELER L1 across the three split settings. Our model, ScanEZ, improves NLL and NLD and it benchmarks temporal predictions (see NLL_t scores, unavailable for Eyettention).

Training Data	Fine-tune	Testing Data	Model	NLL↓	NLD↓	NLL_t↓
ZuCo 1.0	—	ZuCo 1.0	Eyettention	2.653± 0.020	—	—
CELER L1	—			3.060± 0.026	—	—
CELER L1	ZuCo 1.0			2.613± 0.019	—	—
ZuCo 1.0	—	ZuCo 1.0	**ScanEZ**	**1.098± 0.072**	**0.608± 0.013**	**1.461± 0.035**
CELER L1	—			**0.829± 0.057**	**0.718± 0.008**	**1.214± 0.018**
CELER L1	ZuCo 1.0			**0.548± 0.069**	**0.690± 0.018**	**1.156± 0.030**

Table 4: Cross-dataset results following the training-testing set combinations of (Deng et al., 2023). Our model demonstrates better transfer performance than Eyettention based on NLL. We further include NLD and NLL_t scores, which are not reported in Eyettention.

Task	Model	NLL↓	NLD↓	NLL_t↓	FDA↑	FLA↑
Text	w/o Fine-tuning	3.021 ±0.044	0.952 ±0.004	1.406 ±0.027	0.604 ± 0.006	0.285 ± 0.016
	w/o Pre-training	1.715 ±0.254	0.537±0.065	1.419 ±0.259	0.634 ±0.021	0.376 ±0.050
	ScanEZ	**1.603 ±0.017**	**0.430 ±0.047**	**1.269 ±0.068**	**0.653 ±0.011**	**0.408 ±0.008**
Part	w/o Fine-tuning	3.032 ±0.223	0.951 ±0.015	1.409 ±0.073	0.604 ±0.011	0.286 ±0.024
	w/o Pre-training	1.791 ±0.204	0.517 ±0.104	1.536 ±0.292	0.623 ±0.021	0.378 ±0.045
	ScanEZ	**1.555 ±0.013**	**0.424 ±0.025**	**1.235 ±0.072**	**0.651 ±0.012**	**0.412 ±0.007**
P.T.	w/o Fine-tuning	3.035 ±0.204	0.950 ±0.009	1.403 ±0.012	0.603 ±0.004	0.284 ±0.018
	w/o Pre-training	1.772 ±0.304	0.547 ±0.085	1.431 ±0.150	0.611±0.054	0.381 ±0.054
	ScanEZ	**1.524 ±0.042**	**0.421 ±0.022**	**1.244 ±0.027**	**0.646 ±0.004**	**0.413 ±0.009**

Table 5: Ablation results on CELER L1. Removing pre-training or fine-tuning from ScanEZ degrades performance across all metrics. "w/o Pre-training" uses only human data; "w/o Fine-tuning" uses only E-Z Reader synthetic data.

Task	Model	NLL↓	NLD↓	NLL_t↓	FDA↑	FLA↑
Text	w/o Pre-train	1.331± 0.185	0.780± 0.003	1.164± 0.015	0.793± 0.008	0.502± 0.006
	w/o Fine-tune	1.760± 0.072	0.759± 0.004	1.191± 0.014	0.801± 0.005	**0.588± 0.004**
	ScanEZ	**1.000± 0.012**	**0.589± 0.012**	**1.106± 0.022**	**0.803± 0.004**	0.553± 0.020
Part	w/o Pre-training	1.154± 0.176	0.751± 0.062	1.151± 0.029	0.795± 0.007	0.520± 0.041
	w/o Fine-tuning	1.760± 0.070	0.759± 0.004	1.191± 0.018	0.801± 0.007	**0.588± 0.008**
	ScanEZ	**0.928± 0.037**	**0.585± 0.014**	**1.098± 0.026**	**0.803± 0.006**	0.556± 0.013
P.T.	w/o Pre-training	1.217± 0.212	0.777± 0.004	1.177± 0.066	0.785± 0.012	0.491± 0.023
	w/o Fine-tuning	1.837± 0.025	0.761± 0.005	1.181± 0.011	0.797± 0.003	**0.589± 0.006**
	ScanEZ	**0.996± 0.020**	**0.616± 0.013**	**1.083± 0.033**	**0.804± 0.005**	0.534± 0.012

Table 6: Evaluation on EML. "w/o Pre-training" uses only human data, "w/o Fine-tuning" uses only E-Z Reader synthetic data. ScanEZ is pre-trained on synthetic data and fine-tuned on human data.

Metric	CELER L1	ZuCo 1.0	EML
FDA	92.36	64.46	73.07
FLA	102.0	38.79	53.25

Table 7: The t-scores of the regressions performed for each dataset. All the scores are significant at p-value < 0.001.

Improving Fairness of Large Language Models in Multi-document Summarization

Haoyuan Li[1], Rui Zhang[2], Snigdha Chaturvedi[1]
[1]University of North Carolina at Chapel Hill, [2]Pennsylvania State University
{haoyuanl, snigdha}@cs.unc.edu, { rmz5227}@psu.edu

Abstract

Fairness in multi-document summarization (MDS) is crucial for providing comprehensive views across documents with diverse social attribute values, which can significantly impact decision-making. For example, a summarization system that tends to overrepresent negative reviews of products can mislead customers into disregarding good products. Previous works measure fairness in MDS at two levels: summary-level and corpus-level. While summary-level fairness focuses on individual summaries, corpus-level fairness focuses on a corpus of summaries. Recent methods primarily focus on summary-level fairness. We propose FairPO, a preference tuning method that focuses on both summary-level and corpus-level fairness in MDS. To improve summary-level fairness, we propose to generate preference pairs by perturbing document sets. To improve corpus-level fairness, we propose fairness-aware preference tuning by dynamically adjusting the weights of preference pairs. Our experiments show that FairPO outperforms strong baselines while maintaining the critical qualities of summaries. The code is available at https://github.com/leehaoyuan/coverage_fairness.

1 Introduction

Multi-document summarization (MDS) aims to summarize the salient information from multiple documents about an entity, such as reviews of a product. Each of these documents is generally associated with a *social attributes* such as sentiments in reviews. These documents with different social attribute values e.g. positive sentiment or negative sentiment tend to have diverse information or conflicting opinions. It is crucial that the summary fairly represents conflicting information since it can significantly impact decision-making.

Previous works (Shandilya et al., 2018; Olabisi et al., 2022; Huang et al., 2024) measure fairness in MDS at two levels: summary-level or corpus-level. Summary-level fairness measures how fairly a summary represents documents with different social attribute values. Corpus-level fairness measures how fairly a corpus of summaries as a whole represents different social attribute values.

Recent studies (Zhang et al., 2023; Li et al., 2024) find that modern summarization methods like LLMs struggle with both summary-level and corpus-level fairness. To improve the summary-level fairness, Zhang et al. (2023) prompt LLMs to generate summaries based on the distribution of social attributes among documents. However, it relies on users' prior knowledge of fairness issues and social attributes, limiting its effectiveness in practice. Huang et al. (2024) improve the summary-level fairness of T5 (Raffel et al., 2020) by policy gradient, but their method may not generalize to modern models like LLMs. Furthermore, both methods focus exclusively on summary-level fairness, overlooking the corpus-level fairness.

We propose FairPO (**F**air **P**reference **O**ptimization), a preference tuning (Ziegler et al., 2019) method that focuses on both summary-level and corpus-level fairness of LLMs in MDS. While previous works (Stiennon et al., 2020; Roit et al., 2023) uses preference tuning to improve other qualities of summaries, FairPO is the first to use preference tuning for the fairness in MDS. FairPO is based on Direct Preference Optimization (DPO) (Rafailov et al., 2024). To optimize summary-level fairness, FairPO generates preference pairs given perturbed input document sets by removing a small subset of documents with certain social attribute values. To further improve corpus-level fairness, FairPO performs *fairness-aware preference tuning* by dynamically adjusting the weights of preference pairs.

We conduct an empirical evaluation of FairPO using three LLMs: Llama3.1 (AI@Meta, 2024), Mistral (Jiang et al., 2023), and Gemma2 (Team et al., 2024), on the Amazon (Ni et al., 2019), MITweet (Liu et al., 2023), and SemEval datasets

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1143–1154
July 27 - August 1, 2025

(Mohammad et al., 2016). Our experiments show that FairPO outperforms strong baselines while maintaining other critical qualities of summaries, such as relevance and factuality.

Our contributions are as follows:

- We propose FairPO to improve the fairness of LLMs in MDS;
- We propose to improve summary-level and corpus-level fairness by perturbation-based preference pair generation and fairness-aware preference tuning;
- We perform comprehensive experiments to show the effectiveness of FairPO.

2 Background

In this section, we provide background knowledge on fairness in MDS. Let G denote all document sets in a corpus for MDS. Each document set $D \in G$ contains multiple documents $\{d_1, ..., d_n\}$, where each document d_i is labeled with a social attribute $a_i \in \{1, ..., K\}$. For each document set D, a MDS system is supposed to generate a summary S.

To evaluate fairness in MDS, we use Equal Coverage $EC(D, S)$, a summary-level measure, and Coverage Parity $CP(G)$, a corpus-level measure, proposed by Li et al. (2024). Below, we summarize these concepts as introduced in the original paper.

Equal Coverage examines whether each social attribute value has equal probabilities of being covered by the summary S for a document set D. Specifically, it first defines **coverage probability difference** $c(d_i, S)$ as the difference between the coverage probability for the document d_i, $p(d_i, s)$. It also defines the average coverage probability across all documents, $p(d, s)$. To estimate the coverage probability for the document d_i, $p(d_i, s)$, FairPO estimates the probability $p(d_i, s_j)$ that a document d_i is covered by a summary sentence s_j. Specifically, the probability $p(d_i, s_j)$ is estimated as the maximum entailment probability $p(d_{i,l}, s_j)$ between any document chunk $d_{i,l}$ of the document d_i and the summary sentence s_j using an entailment model:

$$p(d_i, s_j) = \max\{p(d_{i,l}, s_j) | d_{i,l} \in d_i\}, \quad (1)$$

The coverage probability for the document d_i, $p(d_i, s)$, is then estimated as the average of the probability $p(d_i, s_j)$:

$$p(d_i, s) = \frac{1}{|S|} \sum_{s_j \in S} p(d_i, s_j), \quad (2)$$

The average coverage probability, $p(d, s)$, is then calculated by averaging coverage probability, $p(d_i, s)$, across all documents in the document set D. Using these values, Equal Coverage calculates the coverage probability difference $c(d_i, S) = p(d_i, s) - p(d, s)$. Equal Coverage value $EC(D, S)$ is then calculated as the average of the absolute average coverage probability difference $c(d_i, S)$ for documents with each social attribute value:

$$EC(D, S) = \frac{1}{K} \sum_{k=1}^{K} |\mathbb{E}(\{c(d_i, S) | a_i = k\})| \quad (3)$$

A lower $EC(D, S)$ indicates a fairer summary S. To evaluate the fairness of a system, we use the average Equal Coverage value across the corpus G.

Coverage Parity examines whether certain social attribute values are systematically overrepresented or underrepresented across the corpus G. Coverage Parity collects these coverage probability differences $c(d_i, S)$ from all input documents of the dataset G whose social attribute value is k into a set C_k. The coverage Parity value $CP(G)$ is then calculated as the average of the absolute average coverage probability difference $c(d_i, S)$ for documents with each social attribute value:

$$CP(G) = \frac{1}{K} \sum_{k=1}^{K} |\mathbb{E}(C_k)|, \quad (4)$$

A lower $CP(G)$ indicates a fairer system. For more details, please refer to Li et al. (2024).

3 FairPO

In this section, we describe our proposed preference tuning method, FairPO.

3.1 Perturbation-based Preference Pair Generation

In this section, we describe how to generate preference pairs based on perturbation. A preference pair for FairPO contains a chosen summary S_c and a rejected summary S_r for the document set D. Ideally, the chosen and rejected summaries should differ significantly in representing documents with different social attribute values. To this end, FairPO generates summaries for perturbed input document sets, where small subsets ($\alpha\%$) of documents with specific social attribute values are removed.

Specifically, FairPO first generates a summary S for the input document set D and identifies its most

overrepresented, k^+, and underrepresented, k^-, social attribute value. For the completeness of information, FairPO only considers social attribute values that appear in more than $\alpha\%$ of the documents (details in App. A.4). These are determined based on the highest or lowest average coverage probability differences, $\mathbb{E}(\{c(d_i, S)|a_i = k\})$. Then, FairPO generates summary S^+ and S^- for the perturbed input document set where $\alpha\%$ of randomly sampled documents with social attribute value a_i of k^+ and k^- are removed. Among summaries S, S^+, S^-, FairPO selects the summary with the lowest Equal Coverage value, indicating the best summary-level fairness, as the chosen summary S_c. The summary with the highest Equal Coverage value is selected as the rejected summary S_r.

3.2 Fairness-aware Preference Tuning

In this section, we describe fairness-aware preference tuning that optimizes summary-level and corpus-level fairness. To achieve this, FairPO dynamically assigns separate weights for the chosen summary S_c and the rejected summary S_r based on estimated corpus-level fairness during training.

FairPO modifies the DPO objective (more explanations in App. A.3) and introduces separate weights, w_c and w_r, for the chosen summary S_c and rejected summary S_r respectively:

$$\sigma(-m)\beta(w_r log\frac{\pi_\theta(S_r|D)}{\pi_{ref}(S_r|D)} - w_c log\frac{\pi_\theta(S_c|D)}{\pi_{ref}(S_c|D)}) \tag{5}$$

where σ is the sigmoid function, π_θ is the policy model, π_{ref} is the reference model, and m is the reward margin as in DPO:

$$m = \beta log\frac{\pi_\theta(S_c|D)}{\pi_{ref}(S_c|D)} - \beta log\frac{\pi_\theta(S_r|D)}{\pi_{ref}(S_r|D)} \tag{6}$$

The term $\sigma(-m)$ in Eq. 5 serves as a scaling factor and FairPO does not consider its gradient.

FairPO assigns weights w_c and w_r to summaries based on their impact on corpus-level fairness. It assigns high weights w_c to chosen summaries that improve corpus-level fairness by balancing the overrepresentation and underrepresentation of social attribute values. Conversely, it assigns high weights w_r to rejected summaries that hurt corpus-level fairness. To estimate corpus-level fairness, FairPO computes the sum of coverage probability differences for documents with social attribute values of k, $C_k(D, S_*) = \sum_{d\in\{d_i|a_i=k\}} c(d, S_*)$ for each chosen or rejected summary, S_*. A summary S_* is considered overrepresenting or underrepresenting the social attribute value k if the sum of coverage probability differences, $C_k(D, S_*)$, is greater or less than zero respectively. In each training step, FairPO estimates the overrepresentation $O(k)$ of social attribute value k:

$$O(k) = \frac{\sum_{(D,S)\in T_k^+} |C_k(D, S)| \cdot \pi_\theta(S|D)/|S|}{\sum_{(D,S)\in T_k^+} \pi_\theta(S|D)/|S|} \tag{7}$$

where T_k^+ is the set of document sets D and corresponding chosen or rejected summaries that overrepresent social attribute value k ($C_k(D, S_*) > 0$) in recent training steps. Similarly, FairPO estimates the underrepresentation $U(k)$ using the set T_k^- of document sets and summaries that underrepresent social attribute value k ($C_k(D, S_*) < 0$) as Eq. 7.

Using the overrepresentation $O(k)$ and underrepresentation $U(k)$, FairPO assigns weight w_c and w_r. Chosen summaries that help balance overrepresentation $O(k)$ and underrepresentation $U(k)$ receive higher weights and vice versa for rejected summaries. For example, the weight w_c should be higher if a systematically underrepresented social attribute value k ($U(k) > O(k)$) is overrepresented by the chosen summary S_c ($C_k(D, S_c) > 0$). For social attribute value k, FairPO computes an intermediate weight $w_{c,k}$ for the chosen summary S_c:

$$w_{c,k} = \frac{2}{1 + (O(k)/(U(k))^{C_k(D,S_c)/\tau}} \tag{8}$$

where τ is the temperature. The weight w_c for chosen summaries is the average intermediate weight $w_{c,k}$ across all social attribute values. The weight w_r for the rejected summary S_r is computed similarly with the intermediate weight $w_{r,k}$:

$$w_{r,k} = \frac{2}{1 + (U(k)/(O(k))^{C_k(D,S_r)/\tau}} \tag{9}$$

The design ensures that summaries improving corpus-level fairness are prioritized.

	Domain	Soci. Attr.	Soci. Attr. Val.	Doc. Set Size	Doc. Len
Amazon	Review	Sentiment	negative, neutral, positive	8	40
MiTweet	Tweet	Ideology	left, center, right	20	34
SemEval	Tweet	Stance	support, against	30	17

Table 1: Dataset statistics. Doc. Set Size means size of document sets. Doc. Len. means average length of documents.

4 Experiments

In this section, we describe experiments of finetuning models with FairPO.

	Amazon		MITweet		SemEval		Overall	
	$EC \downarrow$	$CP \downarrow$	EC	$CP \downarrow$	$EC \downarrow$	$CP \downarrow$	$\overline{EC} \downarrow$	$\overline{CP} \downarrow$
Llama3.1	7.95	1.89	4.50	0.59	2.98	1.41	5.14	1.30
+DPO	7.23	1.27	4.25	0.47	2.66	1.09	4.72	0.94
+OPTune	**6.70**	0.62	4.33	0.51	2.60	0.95	4.54	0.69
+Prompt	7.42	1.64	4.36	0.45	2.62	**0.29**	4.80	0.79
+Policy G.	7.73	1.88	4.51	0.55	2.97	1.38	5.07	1.27
+FairPO	6.87	**0.42**	**4.24**	**0.42**	**2.49**	0.66	**4.53**	**0.50**
Mistral	8.36	2.83	4.16	0.61	2.83	1.27	5.12	1.57
+DPO	7.20	1.82	**3.55**	**0.34**	2.41	0.93	4.39	1.03
+OPTune	6.85	0.88	3.58	0.51	**2.07**	0.57	4.17	0.65
+Prompt	7.74	1.92	3.97	0.37	2.35	**0.36**	4.68	0.88
+FairPO	**6.32**	**0.46**	3.70	0.40	2.10	0.43	**4.04**	**0.43**
Gemma2	8.32	2.48	4.20	0.60	2.81	0.96	5.11	1.35
+DPO	6.90	0.91	4.04	0.40	2.44	0.56	4.46	0.62
+OPTune	6.84	0.88	3.89	0.57	**2.32**	0.49	4.35	0.65
+Prompt	7.28	1.16	4.33	**0.32**	2.73	0.48	4.78	0.65
+FairPO	**6.18**	**0.44**	**3.76**	0.48	2.50	**0.45**	**4.15**	**0.46**

Table 2: Summary-level fairness (EC) and corpus-level fairness (CP) of summaries generated by different methods. The best performing method is in **bold**. The second-best performing method is underlined. FairPO has the best overall performance.

4.1 Datasets

We experiment on three datasets: Amazon (Ni et al., 2019), MITweet (Liu et al., 2023), SemEval (Mohammad et al., 2016) datasets. Each dataset includes 1000 samples for training, 300 samples for validation, and 300 samples for testing. The division of training, validation, and testing sets is based on stratified sampling of social attribute values and topics. Tab. 1 shows the statistics of these datasets. The summary length is 50 words. Details of preprocessing are in App. A.1.

4.2 Implementation Details

We perform experiments with three LLMs: Llama3.1-8b-Instruct (AI@Meta, 2024) , Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), Gemma-2-9b-it (Team et al., 2024). Each LLM is trained for 2 epochs using LoRA (Hu et al., 2021) with a learning rate of $5e-5$ and batch size of 16. To generate preference pairs, FairPO removes $\alpha = 10\%$ of documents. The temperature τ is 1 on the MITweet dataset, 2 for Mistral and 1 for other LLMs on the Amazon dataset, 3 for Mistral and 2 for other LLMs on the SemEval dataset. All hyperparameters are tuned on the validation set. More details are in App. A.4.

4.3 Automatic Evaluation of FairPO

We compare FairPO with the following baselines: (i) DPO (Rafailov et al., 2024), where the chosen and rejected summaries are selected among three randomly sampled summaries based on EC values like FairPO for a fair comparison; (ii) OPTune (Chen et al., 2024), which selects the chosen and rejected summaries as DPO and weights preference pairs based on EC value differences; (iii) Policy gradients (Lei et al., 2024) and (iv) a prompting method (Zhang et al., 2023). Implementation details of these baselines are in App A.5. For evaluation, we consider summary-level and corpus-level fairness using Equal Coverage (EC) and Coverage Parity (CP) (Li et al., 2024). A lower value is better for these measures. We report the average results on three splittings of training, validation and testing in Tab. 2.We additionally report the results for each splitting in App. A.6. We observe that FairPO outperforms other methods for most LLMs on all datasets and yields the best overall performance for all LLMs. The results show that FairPO improves both summary-level and corpus-level fairness.

4.4 Ablation Study

To validate the effect of perturbation-based preference pair generation and fairness-aware preference tuning, we compare FairPO with its ablated versions. We consider the following ablated versions: (i) (w/o pert.), where the chosen and rejected summaries are selected among three randomly sampled summaries based on Equal coverage values; (ii) (w/o fair.) that performs preference tuning using the DPO objective instead of the fairness-aware preference tuning; (iii) (w/o rew.) that directly assigns weights w_c and w_r in the DPO objective (Eq. 13), which undermines the effectiveness of reward margin (more explanations in App. A.3). Tab.3 reports the results for each dataset, and *Overall* scores, which is the average across all datasets. A lower value indicates better fairness.

From the table, we observe that FairPO yields the best overall performance compared to its ablated versions. The results show the effectiveness of perturbation-based preference pair generation and fairness-aware preference tuning. It also provides empirical evidences for the design choice of objective of FairPO.

4.5 Human Evaluation of FairPO

We perform a human evaluation to compare the fairness of summaries generated by LLMs tuned with DPO and FairPO. For each LLM, we randomly select 10 pairs of summaries generated by the LLM tuned with DPO or FairPO, yielding a total of 30 pairs. Each pair is annotated by three annotators recruited from Amazon Mechanical Turk. Annota-

	Amazon		MITweet		SemEval		Overall	
	$EC\downarrow$	$CP\downarrow$	EC	$CP\downarrow$	$EC\downarrow$	$CP\downarrow$	$\overline{EC}\downarrow$	$\overline{CP}\downarrow$
				Llama3.1				
FariPO	6.57	**0.37**	4.20	**0.26**	2.39	**0.56**	**4.39**	**0.39**
w/o pert.	7.01	0.48	**4.07**	0.34	2.54	0.81	4.54	0.54
w/o fair.	6.70	0.95	4.26	0.31	**2.29**	0.65	4.42	0.64
w/o rew	**6.48**	0.79	4.19	0.27	2.60	0.86	4.42	0.64
				Mistral				
FariPO	**6.98**	**0.89**	**3.56**	0.21	**1.97**	**0.36**	**4.17**	**0.49**
w/o pert.	7.29	1.64	3.81	0.21	2.30	0.26	4.47	0.71
w/o fair.	7.31	1.36	3.57	0.25	2.21	0.66	4.37	0.76
w/o rew	7.05	1.26	3.65	**0.14**	2.06	0.55	4.25	0.65
				Gemma2				
FariPO	**6.09**	**0.33**	**3.84**	0.47	2.53	0.59	**4.15**	0.46
w/o pert.	6.18	0.19	4.17	**0.21**	2.43	0.53	4.26	**0.31**
w/o fair.	6.77	1.11	3.84	0.51	**2.39**	0.59	4.34	0.74
w/o rew	6.89	0.90	3.94	0.40	2.49	**0.44**	4.44	0.58

Table 3: Summary-level fairness (EC) and corpus-level fairness (CP) of summaries generated by ablated versions of FairPO. The best performing method is in **bold**. FairPO has the best overall performance.

	Llama3.1			Mistral			Gemma2		
	flu.↑	rel.↑	fac.↑	flu.↑	rel.↑	fac.↑	flu.↑	rel.↑	fac.↑
DPO	7.56	8.33	2.78	5.11	11.56	11.56	5.11	1.11	8.67
OPTune	1.00	0.44	-6.89	-0.78	6.78	8.89	7.00	11.67	11.67
Prompt	-15.33	-19.22	-24.44	-0.44	-6.00	-5.56	-42.67	-50.78	-51.44
FairPO	5.78	3.11	2.89	2.11	5.33	9.11	11.44	16.11	9.44

Table 4: Pairwise comparison of quality between summaries generate by LLMs before and after tuning. Statistical significant differences ($p < 0.05$) according to paired bootstrap resampling (Koehn, 2004) are underlined. FairPO does not affect summary quality.

tors are asked to read all corresponding documents and select the fairer summary. We choose the Amazon dataset since each document set only contains eight reviews (Tab. 1) and judging the sentiment of an opinion is relatively easy for common users. The Randolph's Kappa (Randolph, 2005) between annotations of three annotators is 0.40, which shows a moderate correlation. The correlation is expected considering the subjectivity of the task. More details are in App. A.2.

Out of 30 pairs, summaries generated by FairPO-tuned LLMs are fairer in 18 pairs and summaries generated by DPO-tuned LLMs are fairer in 9 pairs.The difference is statistically significant ($p < 0.05$) using bootstrap (Koehn, 2004). The results show that FairPO performs better than DPO in improving fairness. We additionally show example summaries generated by FairPO in App. A.7.

4.6 Evaluation of Summary Quality

To evaluate FairPO's impact on summary quality, we compare summaries generated by LLMs before and after tuning to improve fairness. Specifically, for a pair of summaries, we instruct Prometheus 2 (7B) (Kim et al., 2024) to select the better summary in three dimensions: fluency, relevance, and factuality. To mitigate position bias (Huang et al., 2023), we perform the pairwise comparison twice with different orders of summaries and only consider consistent results. Tab. 4 reports the differences between the winning and losing rates of different methods. A positive value indicates summary quality is better compared to original LLMs.

From the table, we observe that the quaility of summaries generated by LLMs tuned with FairPO is comparable with summaries generated by original LLMs. Contrarily, prompting significantly hurt the quality of summaries. The results show that FairPO improves the fairness of summaries while maintaining their quality.

5 Conclusion

We propose FairPO, a preference tuning method that optimizes summary-level fairness and corpus-level fairness in MDS. Specifically, FairPO generates preference pairs using perturbed document sets to improve summary-level fairness and performs fairness-aware preference tuning to improve corpus-level fairness. Our experiments show that FairPO outperforms strong baselines while maintaining critical qualities of summaries.

6 Limitation

Our experiments demonstrate FairPO's effectiveness in improving both summary-level and corpus-level fairness of summaries within individual domains. While this work focuses on optimizing fairness within a single domain, extending FairPO to improve fairness simultaneously across multiple domains with diverse social attributes presents a promising future direction. Besides, FairPO currently selects the two summaries with the largest fairness differences among the three generated summaries for preference tuning, following commonly used practices of DPO. Exploring approaches to utilize all three summaries generated by FairPO can be another interesting future direction.

7 Acknowledgment

This work was supported by NSF grant DRL-2112635 and 2338418.

8 Ethical Consideration

The datasets we use are all publicly available. We do not annotate any data on our own. All the models used in this paper are publicly accessible. The inference and finetuning of models are performed on one Nvidia A6000 or Nvidia A100 GPU.

We perform human evaluation experiments on Amazon Mechanical Turk. The annotators were compensated at a rate of $20 per hour. During the evaluation, human annotators were not exposed to any sensitive or explicit content.

References

AI@Meta. 2024. Llama 3 model card.

Lichang Chen, Jiuhai Chen, Chenxi Liu, John Kirchenbauer, Davit Soselia, Chen Zhu, Tom Goldstein, Tianyi Zhou, and Heng Huang. 2024. Optune: Efficient online preference tuning. *arXiv preprint arXiv:2406.07657*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Kung-Hsiang Huang, Philippe Laban, Alexander R Fabbri, Prafulla Kumar Choubey, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2023. Embrace divergence for richer insights: A multi-document summarization benchmark and a case study on summarizing diverse information from news articles. *arXiv preprint arXiv:2309.09369*.

Nannan Huang, Haytham Fayek, and Xiuzhen Zhang. 2024. Bias in opinion summarisation from pretraining to adaptation: A case study in political bias. *arXiv preprint arXiv:2402.00322*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *Preprint*, arXiv:2405.01535.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Yuanyuan Lei, Kaiqiang Song, Sangwoo Cho, Xiaoyang Wang, Ruihong Huang, and Dong Yu. 2024. Polarity calibration for opinion summarization. *arXiv preprint arXiv:2404.01706*.

Haoyuan Li, Yusen Zhang, Rui Zhang, and Snigdha Chaturvedi. 2024. Coverage-based fairness in multi-document summarization. *Preprint*, arXiv:2412.08795.

Songtao Liu, Ziling Luo, Minghua Xu, LiXiao Wei, Ziyao Wei, Han Yu, Wei Xiang, and Bang Wang. 2023. Ideology takes multiple looks: A high-quality dataset for multifaceted ideology detection. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.

Olubusayo Olabisi, Aaron Hudson, Antonie Jetter, and Ameeta Agrawal. 2022. Analyzing the dialect diversity in multi-document summaries. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6208–6221, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Justus J Randolph. 2005. Free-marginal multirater kappa (multirater k [free]): An alternative to fleiss' fixed-marginal multirater kappa. *Online submission*.

Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Leonard Hussenot, Orgad Keller, Nikola Momchev, Sabela Ramos Garea, Piotr Stanczyk, Nino Vieillard, Olivier Bachem, Gal Elidan, Avinatan Hassidim, Olivier Pietquin, and Idan Szpektor. 2023. Factually consistent summarization via reinforcement learning with textual entailment feedback. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6252–6272, Toronto, Canada. Association for Computational Linguistics.

Anurag Shandilya, Kripabandhu Ghosh, and Saptarshi Ghosh. 2018. Fairness of extractive text summarization. In *Companion Proceedings of the The Web Conference 2018*, pages 97–98.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Yusen Zhang, Nan Zhang, Yixin Liu, Alexander Fabbri, Junru Liu, Ryo Kamoi, Xiaoxin Lu, Caiming Xiong, Jieyu Zhao, Dragomir Radev, et al. 2023. Fair abstractive summarization of diverse perspectives. *arXiv preprint arXiv:2311.07884*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Appendix

A.1 Datasets

In this section, we describe how we preprocess the datasets.

Amazon (Ni et al., 2019) consists of reviews with labels of their ratings of different products. We filter out reviews that are non-English or without ratings. We obtain the social attribute of each review based on its rating provided in the dataset. The social attribute of a review will be positive if its rating is 4 or 5, neutral if its rating is 3, and negative if its rating is 1 or 2. To construct training, validation and testing sets, we perform stratified sampling based on the distribution of social attribute values among document sets for each set. Therefore, each set has equal proportions of document sets D dominated by each social attribute values. We sample 1000 products and their corresponding reviews for training, 300 products for validation, and 300 products for testing.

MITTweet (Liu et al., 2023) consists of tweets with labels of political ideologies on different facets about different topics. The social attribute of a tweet will be left if it is left on most facets, right if it is right on most facets, otherwise neutral. First, we evenly divide all tweets of each topic into two parts so that the distribution of topics is the same between two parts. For each part, we cluster tweets about the same topic based on their TFIDF similarity into clusters. We then divide these clusters into input document sets of 20 tweets about the same topic. We generate 1000 input document sets for training from the first part of the tweets. Similarly, we generate 300 input document sets for validation and 300 input document sets for testing from the second part of the tweets. When generating input document sets of training, validation, and testing sets, we also perform stratified sampling based on the distribution of social attribute values so that each set has equal proportions of document sets D dominated by each social attribute value.

Tweet Stance (Mohammad et al., 2016) consists of tweets with labels of stance toward a target phrase such as Climate Change or Hillary Clinton. First, we evenly divide all tweets of each topic into two parts so that the distribution of target phrase is the same between two parts. We cluster tweets about the same target phrase based on their TFIDF similarity into clusters. We then divide these clusters into input document sets of 30 tweets about the same target phrase. We generate 1000 input document sets for training from the first part of the tweets. Similarly, we generate 300 input document sets for validation and 300 input document sets for testing from the second part of the tweets. When generating input document sets of training, validation, and testing sets, we also perform stratified sampling based on the distribution of social attribute values so that each set has equal proportions of document sets D dominated by each social attribute value.

A.2 Human Evaluation

We perform a human evaluation to compare the fairness of summaries generated by LLMs tuned with DPO and FairPO. For each LLM, we randomly se-

lect 10 pairs of summaries generated by the LLM tuned with DPO or FairPO, yielding a total of 30 pairs. To further simplify the evaluation, we consider document sets with only negative and positive reviews. Each pair is annotated by three annotators recruited from Amazon Mechanical Turk. The annotators should be from English-speaking countries and have HIT Approval Rates greater than 98%. For each pair, annotators are first asked to read corresponding reviews and unique opinions automatically extracted by GPT-4o-mini (Ouyang et al., 2022). They then evaluate whether each summary reflects these opinions and classify the summary as leaning negative, fair, or leaning positive. Eventually, they are asked to select the fairer summary in each pair. The interface of human evaluation is shown in Fig. 1.

A.3 Relation between FairPO and DPO

The FairPO objective (Eq. 5) is motivated by the derivate of the DPO objective with respect to the model parameters θ:

$$\sigma(-m)\beta(\pi_\theta(S_r|D)^{-1}\frac{\partial \pi_\theta(S_r|D)}{\partial \theta} - \pi_\theta(S_c|D)^{-1}\frac{\partial \pi_\theta(S_c|D)}{\partial \theta}) \tag{10}$$

where σ is the sigmoid function, π_θ is the policy model, π_{ref} is the reference model, and m is the reward margin in DPO:

$$\beta log\frac{\pi_\theta(S_c|D)}{\pi_{ref}(S_c|D)} - \beta log\frac{\pi_\theta(S_r|D)}{\pi_{ref}(S_r|D)} \tag{11}$$

The reward margin m can be viewed as a measure of the model's ability to distinguish between the chosen summary S_c and the rejected summary S_r. A larger value of m indicates that the model is already proficient at differentiating S_c from S_r. Consequently, DPO assigns lower weights, $\sigma(-m)$, to chosen and rejected summaries where the model is confident in their differences and higher weights to chosen and rejected summaries where the differences is more challenging. The term $\sigma(-m)$ can help the model focuses more on difficult cases.

The objective of FairPO is designed so that chosen and rejected summaries have separate weight while preserving the effect of the term $\sigma(-m)$ in Eq.10. The derivative of FariPO objective with respect to the model parameters θ is as follows:

$$\sigma(-m)\beta(w_r\pi_\theta(S_r|D)^{-1}\frac{\partial \pi_\theta(S_r|D)}{\partial \theta} - w_c\pi_\theta(S_c|D)^{-1}\frac{\partial \pi_\theta(S_c|D)}{\partial \theta}) \tag{12}$$

Comparing with the derivative of DPO objective (Eq. 10), the term $\sigma(-m)$ remains consistent in the derivative of FairPO objective.

Suppose we directly add seperate weights w_c and w_r for chosen and rejected summaries to DPO objective. The corresponding objective is as follows:

$$-log\sigma(\beta w_c log\frac{\pi_\theta(S_c|D)}{\pi_{ref}(S_c|D)} - \beta w_r log\frac{\pi_\theta(S_r|D)}{\pi_{ref}(S_r|D)}) \tag{13}$$

The corresponding derivative is as follows:

$$\sigma(-m')\beta(w_r\pi_\theta(S_r|D)^{-1}\frac{\partial \pi_\theta(S_r|D)}{\partial \theta} - w_c\pi_\theta(S_c|D)^{-1}\frac{\partial \pi_\theta(S_c|D)}{\partial \theta}) \tag{14}$$

where m' is a weighted reward margin:

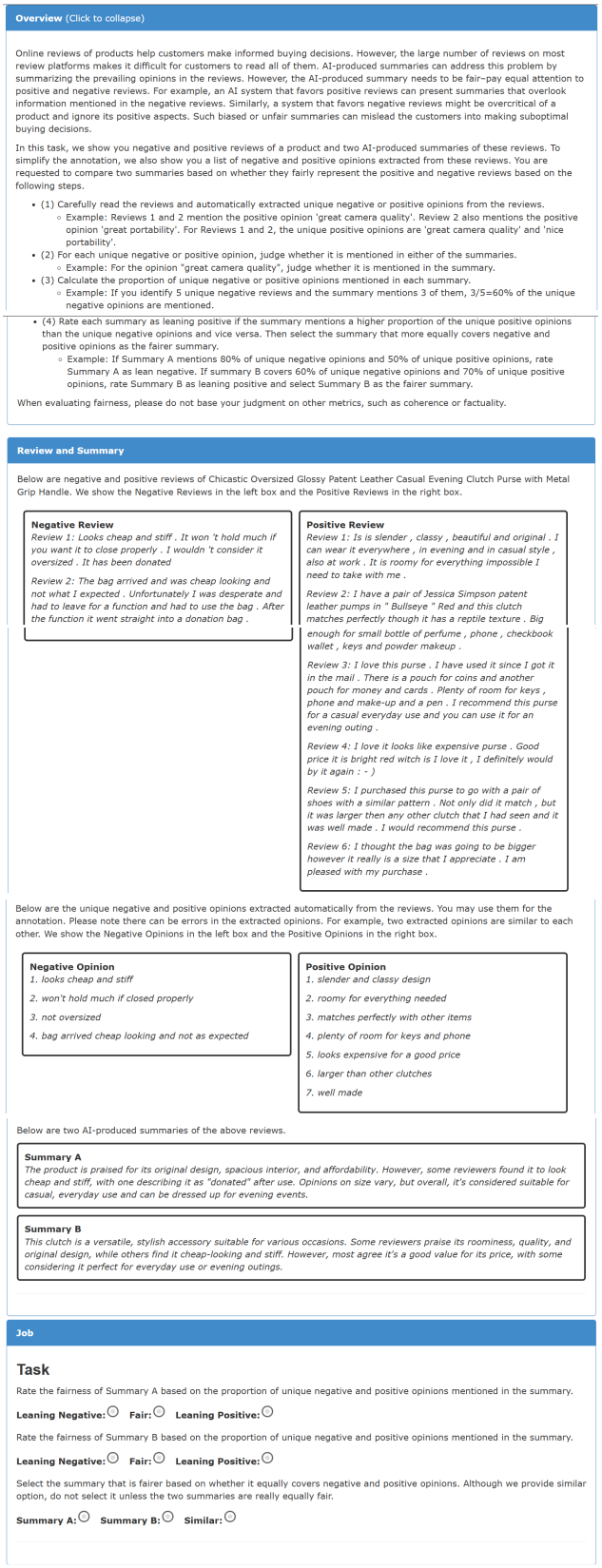
$$\beta w_c log\frac{\pi_\theta(S_c|D)}{\pi_{ref}(S_c|D)} - \beta w_r log\frac{\pi_\theta(S_r|D)}{\pi_{ref}(S_r|D)} \tag{15}$$

Comparing with m, m' is less effective as a measure of the model's ability to distinguish between the chosen summary S_c and the rejected summary S_r since the term $log\frac{\pi_\theta(S_c|D)}{\pi_{ref}(S_c|D)}$ and $log\frac{\pi_\theta(S_r|D)}{\pi_{ref}(S_r|D)}$ have different weights. We additionally provide empirical evidences in App.4.4.

A.4 Implementation Details

To reduce training cost, we perform LoRA (Hu et al., 2021) tuning. Specifically, the rank for LoRA tuning is 16 and the scaling factor is also 16. All models are quantized in 8-bit to additionally reducing training cost.

When performing perturbation on each document set to generate preference pairs, we observe that certain social attribute values are extremely rare in some document sets. If FairPO removes α percent of documents with these rare social attribute values, those social attribute values will disappear entirely from the document set. Therefore, when performing perturbation, we only consider social attribute values that appear in more than α percent of the documents. In the most extreme case, if only one social attribute value meets this requirement, FairPO will sample different subsets of α percent of documents with that social attribute value. By doing this, we assure the completeness of social attribute values after perturbation.



Overview (Click to collapse)

Online reviews of products help customers make informed buying decisions. However, the large number of reviews on most review platforms makes it difficult for customers to read all of them. AI-produced summaries can address this problem by summarizing the prevailing opinions in the reviews. However, the AI-produced summary needs to be fair–pay equal attention to positive and negative reviews. For example, an AI system that favors positive reviews can present summaries that overlook information mentioned in the negative reviews. Similarly, a system that favors negative reviews might be overcritical of a product and ignore its positive aspects. Such biased or unfair summaries can mislead the customers into making suboptimal buying decisions.

In this task, we show you negative and positive reviews of a product and two AI-produced summaries of these reviews. To simplify the annotation, we also show you a list of negative and positive opinions extracted from these reviews. You are requested to compare two summaries based on whether they fairly represent the positive and negative reviews based on the following steps.

- (1) Carefully read the reviews and automatically extracted unique negative or positive opinions from the reviews.
  - Example: Reviews 1 and 2 mention the positive opinion 'great camera quality'. Review 2 also mentions the positive opinion 'great portability'. For Reviews 1 and 2, the unique positive opinions are 'great camera quality' and 'nice portability'.
- (2) For each unique negative or positive opinion, judge whether it is mentioned in either of the summaries.
  - Example: For the opinion "great camera quality", judge whether it is mentioned in the summary.
- (3) Calculate the proportion of unique negative or positive opinions mentioned in each summary.
  - Example: If you identify 5 unique negative reviews and the summary mentions 3 of them, 3/5=60% of the unique negative opinions are mentioned.
- (4) Rate each summary as leaning positive if the summary mentions a higher proportion of the unique positive opinions than the unique negative opinions and vice versa. Then select the summary that more equally covers negative and positive opinions as the fairer summary.
  - Example: If Summary A mentions 80% of unique negative opinions and 50% of unique positive opinions, rate Summary A as lean negative. If summary B covers 60% of unique negative opinions and 70% of unique positive opinions, rate Summary B as leaning positive and select Summary B as the fairer summary.

When evaluating fairness, please do not base your judgment on other metrics, such as coherence or factuality.

Review and Summary

Below are negative and positive reviews of Chicastic Oversized Glossy Patent Leather Casual Evening Clutch Purse with Metal Grip Handle. We show the Negative Reviews in the left box and the Positive Reviews in the right box.

Negative Review

Review 1: Looks cheap and stiff . It won 't hold much if you want it to close properly . I wouldn 't consider it oversized . It has been donated

Review 2: The bag arrived and was cheap looking and not what I expected . Unfortunately I was desperate and had to leave for a function and had to use the bag . After the function it went straight into a donation bag .

Positive Review

Review 1: Is is slender , classy , beautiful and original . I can wear it everywhere , in evening and in casual style , also at work . It is roomy for everything impossible I need to take with me .

Review 2: I have a pair of Jessica Simpson patent leather pumps in " Bullseye " Red and this clutch matches perfectly though it has a reptile texture . Big enough for small bottle of perfume , phone , checkbook wallet , keys and powder makeup .

Review 3: I love this purse . I have used it since I got it in the mail . There is a pouch for coins and another pouch for money and cards . Plenty of room for keys , phone and make-up and a pen . I recommend this purse for a casual everyday use and you can use it for an evening outing .

Review 4: I love it looks like expensive purse . Good price it is bright red witch is I love it , I definitely would by it again : -)

Review 5: I purchased this purse to go with a pair of shoes with a similar pattern . Not only did it match , but it was larger then any other clutch that I had seen and it was well made . I would recommend this purse .

Review 6: I thought the bag was going to be bigger however it really is a size that I appreciate . I am pleased with my purchase .

Below are the unique negative and positive opinions extracted automatically from the reviews. You may use them for the annotation. Please note there can be errors in the extracted opinions. For example, two extracted opinions are similar to each other. We show the Negative Opinions in the left box and the Positive Opinions in the right box.

Negative Opinion

1. looks cheap and stiff

2. won't hold much if closed properly

3. not oversized

4. bag arrived cheap looking and not as expected

Positive Opinion

1. slender and classy design

2. roomy for everything needed

3. matches perfectly with other items

4. plenty of room for keys and phone

5. looks expensive for a good price

6. larger than other clutches

7. well made

Below are two AI-produced summaries of the above reviews.

Summary A

The product is praised for its original design, spacious interior, and affordability. However, some reviewers found it to look cheap and stiff, with one describing it as "donated" after use. Opinions on size vary, but overall, it's considered suitable for casual, everyday use and can be dressed up for evening events.

Summary B

This clutch is a versatile, stylish accessory suitable for various occasions. Some reviewers praise its roominess, quality, and original design, while others find it cheap-looking and stiff. However, most agree it's a good value for its price, with some considering it perfect for everyday use or evening outings.

Job

Task

Rate the fairness of Summary A based on the proportion of unique negative and positive opinions mentioned in the summary.

Leaning Negative: **Fair:** **Leaning Positive:**

Rate the fairness of Summary B based on the proportion of unique negative and positive opinions mentioned in the summary.

Leaning Negative: **Fair:** **Leaning Positive:**

Select the summary that is fairer based on whether it equally covers negative and positive opinions. Although we provide similar option, do not select it unless the two summaries are really equally fair.

Summary A: **Summary B:** **Similar:**

Figure 1: Interface for Human Evaluation

Below is a list of product reviews:

1.This is a card reader that does everything I needed it to . My adapters for the micro SD cards were defective so I have no complaints only praise . It reads any Compact Flash , Memory Stick , SD , and XD cards . Well that is all I wanted to say except this is a great product overall , and thank you .

2.The pins in the CF slot are very flimsy and get bent out of alignment easily , making it impossible to insert the card (until you perform delicate surgery on the pins with small tweezers) . Do not buy this product if you will ever use the CompactFlash slot . It will just lead to frustration .

3.So far I only use this for SM and SD cards , but it installed (USB) quickly , easily and reads the cards I need read .

4.Initially it worked great but after the 5th time it stopped working . It also helped fry my SD-card will all my pictures and video clips . Not happy at all with this product .

5.Reads 64 cards is quite deceiving . It only reads four types of cards made by 64 different manufacturers . Also , the connector port is difficult to plug in .

6.good product , reads quite fast. only issue is that the card reader does not have a satisfying ' click ' when the card is inserted. you kinda have to stick the card in the slot and hope it is lodged properly .

7.I can get it to read SD cards , but I bought it to read my CF 's and it won 't read a single one . My experience is in line with others . Go check out similar reviews on newegg.com.

8.The card reader comes in retail packaging and totally lacks instructions on how best to put 68 types of cards into 4 slots . It did read an SD card successfully . The micro usb plug on the usb cord broke after 1 use .

Please write a single summary around 50 words for all the above reviews.

Figure 2: Summarization prompt for the Amazon Dataset.

We prompt these LLMs to generate summaries for the input document sets of different datasets. The prompt are tuned so that the average length of generated summaries are 50 words. We show the summarization prompts for the Amazon dataset in Fig. 2. The temperature for generation is 0.6 for all LLMs.

The set T_k^+ in Eq.7 is updated so that recent training steps have higher impacts. Specifically, at the end of each training step, the impacts of all the samples already in the set T_k^+ are reduced with a discount factor γ. Then, all the samples that over-represents social attribute value k ($C_k(D, S_*)$>0) in current training steps are added to the set T_k^+. The discount factor γ is 0.75 for Llama3.1 and 0.5 for other LLMs.

The goal of the exponent, $C_k(D, S_*)$, of $O(k)/(U(k)$ or $U(k)/(O(k)$ in Eq. 8 and Eq. 9 is to adjust the weights w_c and w_r such that it more deviates from 1 as $C_k(D, S_*)$ more deviates from 0. Therefore, FairPO does not directly use the raw value of the sum of coverage probability differences $C_k(D, S_*)$ as the exponent. Instead, FairPO separately normalizes $C_k(D, S_*)$ among all training samples where $C_k(D, S_*)$ is greater than zero or less than zero.

A.5 Implementation of Baseline

We implement the policy gradient method proposed by Lei et al. (2024) as a baseline. In the original implementation, there is a loss that maximize the probability for reference summary in addition to the policy gradients. Since datasets used in this paper do not contain reference summary, we only consider the policy gradients. Besides, for a fair comparison with other methods, we implement the policy gradient method in an offline setting. The learning rate for the policy gradeint is $1e-6$ following the original paper. We only implement the policy gradient method for Llama3.1 since the training is very unstable even if we lower the learning rate to $1e-9$ for Mistral and Gemma2. For OPTune and DPO, they use the same hyperparameters as FairPO.

	Amazon		MITweet		SemEval		Overall	
	$EC\downarrow$	$CP\downarrow$	EC	$CP\downarrow$	$EC\downarrow$	$CP\downarrow$	$EC\downarrow$	$CP\downarrow$
Llama3.1	7.90	1.92	4.43	0.26	2.94	1.33	5.09	1.17
+DPO	6.87	1.04	**4.03**	0.31	2.55	0.91	4.49	0.75
+OPTune	6.58	0.75	4.22	**0.23**	2.50	0.81	4.43	0.60
+Prompt	7.71	1.84	4.33	0.38	2.53	**0.26**	4.86	0.83
+Policy G.	7.71	2.10	4.46	0.31	2.95	1.32	5.04	1.24
+FairPO	**6.57**	**0.37**	4.20	0.26	**2.39**	0.56	**4.39**	**0.39**
Mistral	8.18	2.98	3.98	0.42	2.67	1.07	4.94	1.49
+DPO	7.17	1.55	3.60	0.28	2.21	0.64	4.33	0.82
+OPTune	7.48	1.56	3.60	0.25	2.00	0.67	4.36	0.83
+Prompt	7.67	1.93	4.02	0.23	2.38	0.38	4.69	0.85
+FairPO	**6.98**	**0.89**	**3.56**	**0.21**	**1.97**	**0.36**	**4.17**	**0.49**
Gemma2	8.44	2.75	4.17	0.34	2.74	0.91	5.12	1.33
+DPO	6.87	1.04	4.04	0.29	2.42	0.70	4.44	0.68
+OPTune	6.90	1.15	3.86	0.45	**2.40**	0.65	4.39	0.75
+Prompt	7.21	1.13	4.28	**0.24**	2.62	**0.30**	4.70	0.56
+FairPO	**6.09**	**0.33**	**3.84**	0.47	2.53	0.59	**4.15**	**0.46**

Table 5: Summary-level fairness (EC) and corpus-level fairness (CP) of summaries generated by different methods on the first splitting of training, validation and testing.. The best performing method is in **bold**. The second-best performing method is underlined. FairPO has the best overall performance on the first splitting.

A.6 Results using Different Dataset Splitting

To validate the stability of FairPO on three different splittings of datasets, we generate the training, vali-

	Amazon		MITweet		SemEval		Overall	
	$EC \downarrow$	$CP \downarrow$	EC	$CP \downarrow$	$EC \downarrow$	$CP \downarrow$	$\overline{EC} \downarrow$	$\overline{CP} \downarrow$
Llama3.1	7.90	2.05	4.50	0.63	2.90	1.41	5.10	1.36
+DPO	7.27	1.37	4.30	0.37	2.70	1.12	4.76	0.95
+OPTune	**6.92**	**0.40**	4.30	0.52	2.82	1.00	4.68	0.64
+Prompt	7.28	1.67	4.41	0.44	2.74	**0.51**	4.81	0.87
+Policy G.	7.75	1.85	4.47	0.48	2.80	1.30	5.02	1.21
+FairPO	6.96	0.44	**4.26**	**0.29**	**2.69**	0.59	**4.64**	**0.44**
Mistral	8.60	2.74	4.18	0.73	2.91	1.28	5.23	1.58
+DPO	7.24	1.79	**3.39**	**0.26**	2.70	1.15	4.44	1.07
+OPTune	6.59	0.52	**3.57**	0.53	**2.04**	0.58	4.07	0.54
+Prompt	7.90	1.76	3.74	0.51	2.43	0.52	4.69	0.93
+FairPO	**6.06**	**0.11**	3.83	0.39	2.13	**0.33**	**4.01**	**0.28**
Gemma2	8.31	2.33	4.30	0.80	2.97	1.03	5.19	1.38
+DPO	7.04	0.98	4.07	0.44	2.43	0.48	4.51	0.63
+OPTune	6.91	0.56	3.94	0.86	**2.35**	0.56	4.40	0.66
+Prompt	7.33	1.26	4.49	**0.44**	2.91	0.85	4.91	0.85
+FairPO	**6.09**	**0.44**	**3.82**	0.65	2.70	**0.32**	**4.20**	**0.47**

Table 6: Summary-level fairness (EC) and corpus-level fairness (CP) of summaries generated by different methods on the second splitting of training, validation and testing. The best performing method is in **bold**. The second-best performing method is underlined. FairPO has the best overall performance on the second splitting.

	Amazon		MITweet		SemEval		Overall	
	$EC \downarrow$	$CP \downarrow$	EC	$CP \downarrow$	$EC \downarrow$	$CP \downarrow$	$\overline{EC} \downarrow$	$\overline{CP} \downarrow$
Llama3.1	8.06	1.70	4.57	0.87	3.11	1.51	5.25	1.36
+DPO	7.55	1.39	4.43	0.74	2.73	1.24	4.90	1.12
+OPTune	**6.61**	0.72	4.47	0.79	2.48	1.03	**4.52**	0.85
+Prompt	7.26	1.42	4.35	**0.53**	2.59	**0.11**	4.74	0.69
+Policy G.	7.72	1.69	4.60	0.85	3.16	1.53	5.16	1.36
+FairPO	7.07	**0.44**	**4.25**	0.71	**2.38**	0.82	4.57	**0.66**
Mistral	8.29	2.76	4.32	0.67	2.90	1.46	5.17	1.63
+DPO	7.20	2.11	3.65	0.48	2.32	0.99	4.39	1.19
+OPTune	6.47	0.55	**3.58**	0.76	**2.18**	0.46	4.08	0.59
+Prompt	7.66	2.07	4.14	**0.36**	2.23	**0.19**	4.68	0.87
+FairPO	**5.92**	**0.38**	3.71	0.61	2.21	0.59	**3.95**	**0.53**
Gemma2	8.21	2.36	4.14	0.67	2.72	0.96	5.02	1.33
+DPO	6.80	0.70	4.01	0.49	2.46	0.49	4.42	0.56
+OPTune	6.72	0.94	3.86	0.41	**2.21**	**0.25**	4.27	0.53
+Prompt	7.29	1.09	4.21	**0.28**	2.66	0.28	4.72	0.55
+FairPO	**6.35**	**0.56**	**3.64**	0.32	2.29	0.43	**4.09**	**0.44**

Table 7: Summary-level fairness (EC) and corpus-level fairness (CP) of summaries generated by different methods on the third splitting of training, validation and testing. The best performing method is in **bold**. The second-best performing method is underlined. FairPO has the best overall performance on the third splitting.

dation and testing set using different random seeds and run the automatic evaluation on these splittings. The results for each splitting are shown in Tab. 5, 6, 7 respectively. From the table, we observe that FairPO still shows the best overall performance, which shows the stability of FairPO on different splittings of datasets.

A.7 Qualiative Example

We show sample summaries generated by LLMs tuned with DPO and FairPO on the Amazon dataset in Fig. 3. From the figure, we observe that summaries generated by LLMs tuned FairPO tend to more balancely present negative and positive information.

Llama3.1	Mistral	Gemma2
DPO: The product reviews are mixed, with some customers expressing satisfaction with the quality and price, while others find it flimsy, useless, and painful to use. Issues include bending, crookedness, and difficulty with hair removal, leading to disappointment and a recommendation to avoid the product.	**DPO:** This tablet receives generally positive reviews, praised for its portability, picture quality, speed, and features like USB and HDMI ports. Some users found the battery life lacking, while one experienced a defective product. Overall, it's considered a good value for the price and user-friendly, with a few noting its similarity to Android phones.	**DPO:** This plush bed topper is generally well-received for its comfort and ability to stay in place thanks to its straps. Many users praise its cloud-like softness and how it elevates their sleep quality. However, some users found it loses its fluffiness over time and requires manual plumping.
FairPO: The product reviews are mixed, with some users praising its availability and quality, while others criticize its flimsiness, uselessness, and high price. Some users find the tips too long, weak, or hard to use, while others appreciate its effectiveness, but note that it may not be suitable for all purposes.	**FairPO:** This Toshiba tablet receives mixed reviews. Users praise its lightweight design, fast speed, great picture quality, and affordability. Some highlight its functionality similar to laptops and its ease of use. However, concerns arise regarding battery life and a negative experience with a damaged, non-functional product.	**FairPO:** This plush bed topper receives rave reviews for its comfort, fluffiness, and stay-in-place straps. Some users find it a dream come true, while others experience flattening over time. It's praised for its luxurious feel, comparable to a feather bed without the prickliness. While some find it ideal for air mattresses and adding height, others note back pain issues.

Figure 3: Sample summaries generated by DPO and FairPO.

Should I Believe in What Medical AI Says? A Chinese Benchmark for Medication Based on Knowledge and Reasoning

Yue Wu, Yangmin Huang*, Qianyun Du*, Lixian Lai, Zhiyang He, Jiaxue Hu, Xiaodong Tao
Xunfei Healthcare Technology Co., Ltd.
{yuewu21, ymhuang9, qydu, lxlai2, zyhe, jxhu2, xdtao}@iflytek.com

Abstract

Large language models (LLMs) show potential in healthcare but often generate hallucinations, especially when handling unfamiliar information. In medication, a systematic benchmark to evaluate model capabilities is lacking, which is critical given the high-risk nature of medical information. This paper introduces a Chinese benchmark aimed at assessing models in medication tasks, focusing on knowledge and reasoning across six datasets: indication, dosage and administration, contraindicated population, mechanisms of action, drug recommendation, and drug interaction. We evaluate eight closed-source and five open-source models to identify knowledge boundaries, providing the first systematic analysis of limitations and risks in proprietary medical models.

1 Introduction

Large language models (LLMs) have made significant strides in various domains, including medication, where they provide information and recommendations related to medical treatments (Singhal et al., 2022; Nori et al., 2023). However, a significant challenge remains: these models are prone to generating hallucinations and confidently providing incorrect or incomplete information, especially in cases where they lack adequate knowledge (Stefansson and Johansson, 2021; Shukla et al., 2022). In the context of medication and drug usage, such hallucinations can lead to critical errors, particularly in high-risk situations like identifying contraindicated populations or recommending unsafe drug combinations. Despite the progress made in medical AI, a notable gap remains in the development of systematic benchmarks to evaluate the full range of a model's capabilities in medication applications.

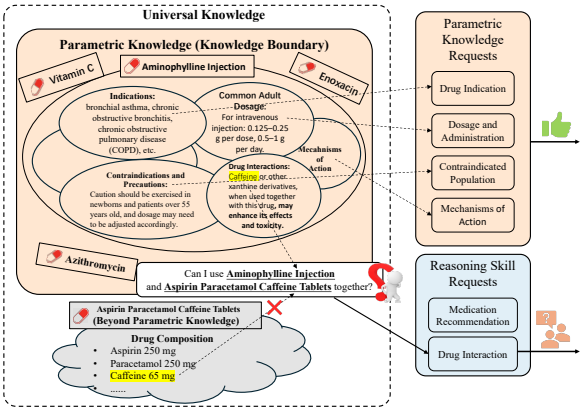


Figure 1: Our benchmark involves four datasets that directly examine model parametric knowledge and two datasets that examine model reasoning ability.

Considering that many existing public benchmarks are constructed by collecting data from online sources and may be susceptible to data leakage risks, we construct a Chinese benchmark, **ChiDrug**, from scratch, based on authoritative drug brochures. This benchmark is specifically designed to evaluate LLMs' knowledge and reasoning capabilities in the medication domain. As shown in Figure 1, our benchmark is structured into two key subdimensions: **parametric knowledge** and **reasoning capability**. We construct six diverse datasets that cover crucial aspects of drug information, including dosage and administration, indication, contraindicated population, mechanisms of action, medication recommendation, and drug interaction.

To evaluate the capabilities of existing models, we apply our benchmark to eight closed-source and five open-source models. Our work also explores various methods for expressing knowledge boundaries, providing insights into the potential risks of overconfident but inaccurate AI-generated responses.

Our contributions include: (1) This benchmark

*Corresponding authors.
Our data and code are available at: https://github.com/LeoCoder33/ChiDrug-benchmark

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1155–1164
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

serves as the first systematic tool for analyzing the capabilities of LLMs in the field of medicine across various dimensions. (2) To the best of our knowledge, this is the first work to systematically conduct knowledge boundary analysis on medical models in the medication domain, providing a comprehensive overview of their performance in real-world medical applications.

2 Related Work

2.1 Chinese Benchmark in Medication

Evaluating the capabilities of Large Language Models (LLMs) in the medical field necessitates specialised benchmarks, particularly when addressing Chinese medical texts. Recent efforts have led to the development of several Chinese-specific medical benchmarks, focusing on various domains such as clinical question answering, knowledge recall, and medication recommendations (Singhal et al., 2023; Liu et al., 2024; Wang et al., 2024; Yue et al., 2024).

MedExpQA (Alonso et al., 2024) proposes a multilingual benchmark evaluating models on medical question-answering tasks, including drug-related and clinical guideline questions. DialMed (He et al., 2022) focuses on dialogue-based medication recommendations, testing models on handling patient symptom queries and drug interactions. However, existing datasets do not have a dedicated benchmark built in the field of medication in Chinese to evaluate the model's ability in this area.

2.2 Abstention in LLMs

The ability of Large Language Models (LLMs) to refrain from providing answers when uncertain—is crucial for enhancing model reliability and safety. Studies have explored various methods to improve this capability (Wen et al., 2024):

Currently, methods to guide models in refusing to answer include: **Calibration-Based Methods**: After the model provides an answer, continue by asking, "Are you sure about your answer?" to verify its confidence (Tian et al., 2023). **Training-Based Methods**: Construct a training set containing both questions the model can answer and those it cannot, training the model to refuse to answer questions with unfamiliar knowledge (Slobodkin et al., 2023; Zhang et al., 2023; Stengel-Eskin et al., 2024). **Consistency-Based Methods**: Perform multiple samplings and calculate the consistency score of the model's responses to assess reliability (Kuhn et al., 2022; Feng et al., 2024a). **Token Probability Methods**: Ensemble the probability of each token generated by the model to determine the uncertainty of the response (Liang et al., 2024; Malinin and Gales, 2021).

3 Dataset

ChiDrug is designed to assess models' parametric knowledge and reasoning ability in handling critical medication-related tasks. Below, we outline the dataset construction process and the verification procedures used to ensure the quality and reliability of the data. The entire benchmark construction process is shown in Figure 2.

3.1 Dataset Construction

We began by collecting official drug brochures for existing medications from the internet[1]. We organized this information into a table that includes details on 8,000 drugs, encompassing their generic names, ingredients, specifications, indications, dosages, contraindications, drug interactions, adverse reactions, and mechanisms of action. This structured dataset served as the foundation for developing questions that evaluate the model's parametric knowledge in four areas: **Indication**, **Dosage and Administration**, **Contraindicated Population**, and **Mechanism of Action**. We extracted the relevant sections from each drug brochure and utilized Spark[2] to generate multiple-choice question stems and answer options. In constructing these questions, we ensured that the incorrect options did not overlap with the correct ones (the left part of Figure 2).

The second step involved constructing questions for **Medication Recommendation** (the middle part of Figure 2). We collected doctor-patient dialogues from the existing DIALMED dataset (He et al., 2022), where Spark transformed these dialogues into question formats, using the doctor's recommended medication as the correct option. To generate distractor options that could confuse the model, we first used Spark to extract the patient's symptoms and demographic information, then searched the drug brochures for medications that treat the same symptoms but are not suitable for the patient's demographic group, thereby creating incorrect options (e.g., "symptom in indication and de-

[1] https://drugs.dxy.cn
[2] https://xinghuo.xfyun.cn

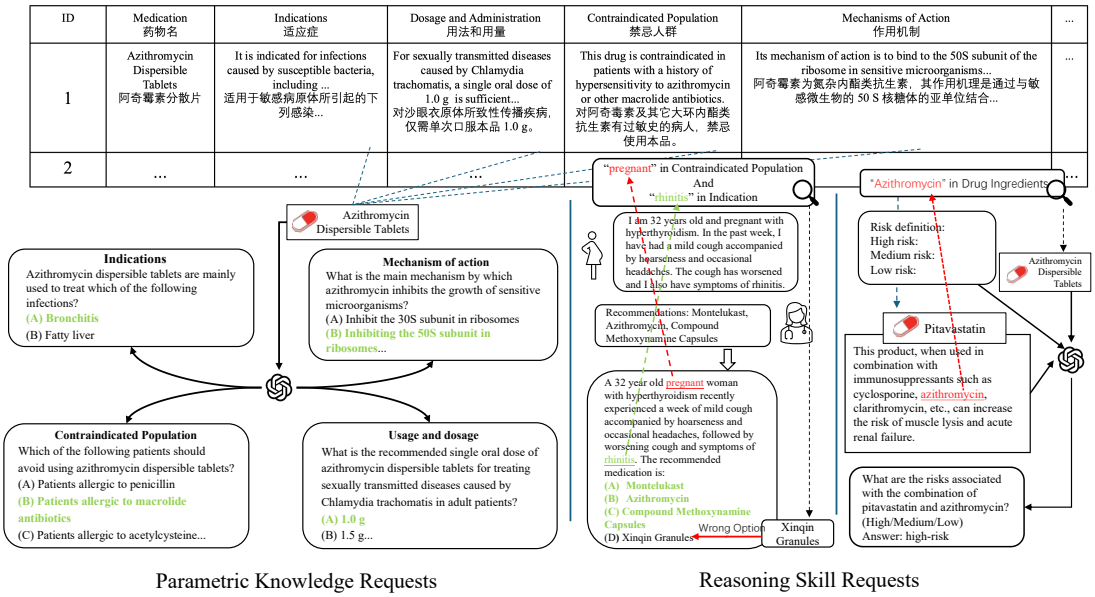


Figure 2: Overview of our benchmark construction process

mographic in contraindicated population").

In the third step, we constructed a dataset for **Drug Interaction** (the right part of Figure 2). First, doctors defined three risk levels for drug interaction (high, medium, and low). We then randomly selected a drug from the brochures and identified its combination guidelines. From there, we extracted the ingredients involved in drug interactions and further searched for medications that contained the same ingredients. Finally, we input the two drugs and the interaction documentation into Spark to generate the appropriate risk level as the correct answer.

3.2 Verification

Since we automatically generated the questions for the dataset, we implemented a double-check process to ensure the questions were reasonable. Each question was tested by three large models (GPT-4[3], Qwen-max[4], ERNIE bot[5]). We gave these models the question, options, and document sources and asked them to check the following: (1) If the question is clear, well-phrased, and free of ambiguity. (2) If the answer is correct. (3) If the answer is unique. A question was considered valid only if all three models agreed it was correct. During dataset construction, we initially generated 7,100 questions. In the LLM verification stage, Spark-generated questions had an acceptance rate of 79.32%, resulting in the removal of 1,468 questions.

Additionally, we hire doctors with licensed qualifications to examine all the datasets we construct (Appendix C).

Ultimately, we construct a benchmark dataset comprising a total of 5,243 samples, encompassing the following categories: Indication (705), Dosage and Administration (651), Contraindicated Population (659), Mechanism of Action (773), Medication Recommendation (838), and Drug Interaction (1,617). More details are provided in Appendix B.

4 Experiment

In this section, we evaluate the performance of large language models (LLMs) on our benchmark. We assess both closed-source and open-source models, using our benchmark to examine their capabilities in handling medication-related queries and their ability to identify knowledge gaps and overconfidence. Table 1 presents the results for the model ability, while the second table focuses on the methods to express the knowledge boundaries in seven different methods.

4.1 Model Performance Evaluation

We selected models with strong Chinese language capabilities, including GPT4o (Hurst et al., 2024), Claude3.5-Sonnet[6], Qwen-max[7], Doubao[8], GLM4

[3] https://chatgpt.com
[4] https://tongyi.aliyun.com/qianwen
[5] https://yiyan.baidu.com
[6] https://claude.ai
[7] https://tongyi.aliyun.com/qianwen
[8] https://www.doubao.com/chat

Close-source Models							
	Dosage and Administration	Indication	Contraindicated Population	Mechanism of Action	Medication Recommendation	Drug Interaction	Avg.
XiaoYi	**81.1**	**77.87**	66.71	92.85	**65.31**	**63.27**	**73.52**
GPT4o	66.41	73.65	69.35	92.13	59.79	59.93	**70.21**
ERNIE	67.64	65.3	57.97	92.76	51.43	38.59	62.28
Qwen-max	69.02	72.13	68.19	93.28	61.22	54.73	69.76
Doubao	71.32	71.24	54.17	92.77	63.25	55.35	68.02
GLM4	71.32	75.71	**71.02**	**94.16**	59.79	54.92	71.15
Claude3.5	54.59	74.53	70.29	89.92	54.06	60.73	67.24
Baichuan4	62.14	69.97	69.24	90.35	52.98	52.81	66.25
Open-source Models							
Bencao	28.92	19.88	12.2	40.71	16.23	**38.28**	26.04
MedGLM	38.92	13.21	8.75	44.86	20.17	34.59	26.75
MedicalGPT	33.51	10.14	3.18	49.41	13.84	30.98	23.51
ChiMedical	33.51	16.04	14.32	38.54	24.71	36.05	27.20
HuatuoGPT2	**55.83**	**47.03**	**18.66**	**77.16**	**25.18**	25.60	**41.58**

Table 1: This table presents the performance of 8 closed-source models and 5 open-source models across various medication-related tasks. Bold indicates the best performance, while underlining denotes the second-best.

	Dosage and Administration		Indication		Contraindicated Population		Mechanism of Action		Medication Recommendation		Drug Interaction		Avg.	
	Precision	A-Acc	Precision	A-Acc	Precision	A-Acc	Precision	A-Acc	Precision	A-Acc	Precision	A-Acc	Precision	A-Acc
Baseline	55.83	55.83	47.03	47.03	18.66	18.66	77.16	77.16	25.18	25.18	25.60	25.60	41.58	41.58
Post-calibration	55.94	**57.91**	47.01	47.62	18.05	18.17	77.21	77.37	25.23	26.12	24.76	25.67	41.37	42.12
IDK	54.26	54.68	47.18	47.18	17.90	17.30	78.68	80.28	24.47	24.66	24.55	26.64	41.17	41.79
LNS	53.99	50.46	50.78	50.85	18.18	23.03	79.97	71.07	28.22	**26.21**	24.94	23.58	42.68	40.87
Probing	66.11	43.90	66.11	51.86	22.83	27.43	85.24	80.01	29.82	19.80	31.31	30.54	50.24	42.26
R-tuning	53.90	57.61	53.66	**56.00**	22.12	**31.00**	81.35	83.00	19.80	12.45	30.34	31.68	43.53	45.29
Self-Consistency	**66.85**	46.20	68.03	48.46	15.24	10.67	**89.37**	78.60	**30.15**	20.84	20.90	30.00	48.42	39.13
Semantic Entropy	64.79	55.32	**70.28**	52.71	**28.11**	26.95	86.24	**83.09**	29.22	24.95	**38.56**	**31.76**	**52.87**	**45.79**

Table 2: This table displays the performance of 7 different methods on the models' ability to detect knowledge boundaries and manage uncertainty.

(GLM et al., 2024), Baichuan4[9], XiaoYi[10], and ERNIE Bot[11], for evaluation of closed-source models. For open-source models, we chose Bencao (Wang et al., 2023), MedGLM (Haochun Wang, 2023), MedicalGPT (Xu, 2023), ChiMedical (Tian et al., 2024), and HuatuoGPT2 (Chen et al., 2024) for evaluation.

The results summarised in Table 1 show that the closed-source models generally outperformed the open-source models across all dimensions, with XiaoYi leading in overall performance, followed closely by GPT4o and ERNIE Bot. In Open-source models, Bencao and MedicalGPT demonstrated lower performance, particularly in complex tasks like Contraindicated Populations and Drug Interactions, while HuatuoGPT2 generally outperformed other models. We will provide a more detailed analysis of each model in the Appendix A.

4.2 Methods for Knowledge Boundary Detection

4.2.1 Task Definition

A formal definition of knowledge boundary detection can be briefly stated as follows:

Input: A model M and a query q.

Output: A response r where

$$r = \begin{cases} M(q), & \text{within parametric knowledge} \\ U, & \text{beyond parametric knowledge} \end{cases} \quad (1)$$

where $M(q)$ is the model's generated answer, and U is an explicit uncertainty expression or abstention (e.g., "I don't know", a confidence score, or an alternative uncertainty marker).

4.2.2 Methods

In this subsection, we apply seven methods to explore their impact on expressing uncertainty or abstention, using HuatuoGPT2 as the backbone.

Post-Calibration (Tian et al., 2023): Enhances model confidence by prompting it to verbalize its certainty after providing an answer.

IDK (I Don't Know): Similar to non-of-the-above (NOTA) in (Feng et al., 2024b), we incorporate an additional "I don't know" option and instruct the model to abstain from answering.

LNS (Malinin and Gales, 2021): Utilizes probabilistic ensemble-based techniques to assess uncertainty in structured prediction tasks, aiding in more reliable outputs.

Probing (Slobodkin et al., 2023): Analyzes internal model representations to understand how they encode information about answerability, helping detect overconfidence and hallucinations.

[9] https://www.baichuan-ai.com
[10] https://chatdr.iflyhealth.com
[11] https://yiyan.baidu.com

R-tuning (Zhang et al., 2023): Instructs models to explicitly state when they lack knowledge, reducing the generation of hallucinated information.

Self-Consistency (Kuhn et al., 2023): Enhances reasoning by generating multiple reasoning paths and selecting the most consistent answer, improving response reliability.

Semantic Entropy (Feng et al., 2024a): Estimates uncertainty in natural language generation by considering linguistic invariances, allowing models to assess the reliability of their outputs better.

4.2.3 Evaluation Metrics

Given the definition as follows:

C = the number of correct answers.

A = the number of total answered questions (excluding abstentions).

A_{correct} = the number of correct abstentions, i.e., questions the model correctly refused to answer because the answer was unknown or uncertain.

N = the total number of questions.

In this experiment, two evaluation metrics are used:

Precision: Measure the proportion of correct answers out of the total predictions made, without abstaining.

$$\text{Precision} = \frac{C}{A} \tag{2}$$

Abstain Accuracy (Feng et al., 2024a): Evaluates the proportion of correct answers and correct abstention due to uncertainty.

$$\text{Abstain-Acc} = \frac{C + A_{\text{correct}}}{N} \tag{3}$$

Results are shown in Table 2. Post-calibration and IDK cannot achieve good results in HuatuoGPT2 due to its weak instruction capabilities. Self-Consistency improved accuracy in complex tasks like Medication Recommendations. Probing refined uncertainty estimations with varying effectiveness. R-tuning reduced hallucinations but sometimes sacrificed performance on complex tasks, while LNS showed mixed results, improving Medication Recommendations but hindering performance on Drug Interaction. Overall, Semantic Entropy has achieved good results in both metrics, and we further analyze the effectiveness of this method on multiple models in Appendix D.

5 Conclusion

We present ChiDrug, a benchmark designed to evaluate LLMs (Large Language Models) in medication-related tasks, with an emphasis on their knowledge and reasoning abilities. Both GLM4 and XiaoYi performed exceptionally well; however, even these advanced models exhibited gaps in drug knowledge. Our work highlights the need for effective methods to align the knowledge boundaries of LLMs, particularly for high-risk tasks.

6 Limitations

This study primarily focuses on Chinese medical texts, which may limit its generalizability. The benchmark doesn't fully capture the complexities of real-world medical decision-making. Additionally, model generalization to new knowledge, handling uncertainty, and reliance on high-quality, up-to-date data are ongoing challenges for AI in healthcare.

7 Ethical considerations

The medication dictionary we constructed is entirely sourced from DingXiangYuan[12], a public medical website. The data on DingXiangYuan was also collected from the China National Medical Products Administration, and the website has a statement allowing non-commercial citations. Documents within our dictionary do not contain private information, so there is no risk of privacy leakage. All the drug information we collect has obtained the national drug approval certificate and is free from copyright issues, in accordance with the regulations of the Chinese government.

References

Iñigo Alonso, Maite Oronoz, and Rodrigo Agerri. 2024. Medexpqa: Multilingual benchmarking of large language models for medical question answering. *Artificial intelligence in medicine*, 155:102938.

Junying Chen, Xidong Wang, Ke Ji, Anningzhe Gao, Feng Jiang, Shunian Chen, Hongbo Zhang, Dingjie Song, Wenya Xie, Chuyi Kong, Jianquan Li, Xiang Wan, Haizhou Li, and Benyou Wang. 2024. Huatuogpt-ii, one-stage training for medical adaption of llms. *Preprint*, arXiv:2311.09774.

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. 2024a. Don't hallucinate, abstain: Identifying llm knowledge gaps via multi-llm collaboration. *arXiv preprint arXiv:2402.00367*.

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. 2024b.

[12] https://drugs.dxy.cn

Don't hallucinate, abstain: Identifying llm knowledge gaps via multi-llm collaboration. *Preprint*, arXiv:2402.00367.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Sendong Zhao Bing Qin Ting Liu Haochun Wang, Chi Liu. 2023. Chatglm-med: chatglm. https://github.com/SCIR-HI/Med-ChatGLM.

Zhenfeng He, Yuqiang Han, Zhenqiu Ouyang, Wei Gao, Hongxu Chen, Guandong Xu, and Jian Wu. 2022. Dialmed: A dataset for dialogue-based medication recommendation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 721–733.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2022. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Xun Liang, Shichao Song, Zifan Zheng, Hanyu Wang, Qingchen Yu, Xunkai Li, Rong-Hua Li, Yi Wang, Zhonghao Wang, Feiyu Xiong, et al. 2024. Internal consistency and self-feedback in large language models: A survey. *arXiv preprint arXiv:2407.14507*.

Mianxin Liu, Jinru Ding, Jie Xu, Weiguo Hu, Xiaoyang Li, Lifeng Zhu, Zhian Bai, Xiaoming Shi, Benyou Wang, Haitao Song, Pengfei Liu, Xiaofan Zhang, Shanshan Wang, Kang Li, Haofen Wang, Tong Ruan, Xuanjing Huang, Xin Sun, and Shaoting Zhang. 2024. Medbench: A comprehensive, standardized, and reliable benchmarking system for evaluating chinese medical large language models. *Preprint*, arXiv:2407.10990.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. 2023. Capabilities of gpt-4 on medical challenge problems. *Preprint*, arXiv:2303.13375.

Abhay Shukla, Paheli Bhattacharya, Soham Poddar, Rajdeep Mukherjee, Kripabandhu Ghosh, Pawan Goyal, and Saptarshi Ghosh. 2022. Legal case document summarization: Extractive and abstractive methods and their evaluation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1048–1064.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, Philip Mansfield, Blaise Aguera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. 2022. Large language models encode clinical knowledge. *Preprint*, arXiv:2212.13138.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. 2023. Towards expert-level medical question answering with large language models. *Preprint*, arXiv:2305.09617.

Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. 2023. The curious case of hallucinatory (un) answerability: Finding truths in the hidden states of over-confident large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3607–3625.

Elis Stefansson and Karl H. Johansson. 2021. Computing complexity-aware plans using kolmogorov complexity. *Preprint*, arXiv:2109.10303.

Elias Stengel-Eskin, Peter Hase, and Mohit Bansal. 2024. Lacie: Listener-aware finetuning for confidence calibration in large language models. *arXiv preprint arXiv:2405.21028*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442.

Yuanhe Tian, Ruyi Gan, Yan Song, Jiaxing Zhang, and Yongdong Zhang. 2024. Chimed-gpt: A chinese medical large language model with full training regime

and better alignment to human preferences. *Preprint*, arXiv:2311.06025.

Haochun Wang, Chi Liu, Nuwa Xi, Zewen Qiang, Sendong Zhao, Bing Qin, and Ting Liu. 2023. Huatuo: Tuning llama model with chinese medical knowledge. *arXiv preprint arXiv:2304.06975*.

Xidong Wang, Guiming Chen, Song Dingjie, Zhang Zhiyi, Zhihong Chen, Qingying Xiao, Junying Chen, Feng Jiang, Jianquan Li, Xiang Wan, Benyou Wang, and Haizhou Li. 2024. CMB: A comprehensive medical benchmark in Chinese. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6184–6205, Mexico City, Mexico. Association for Computational Linguistics.

Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, Bill Howe, and Lucy Lu Wang. 2024. Know your limits: A survey of abstention in large language models. *arXiv preprint arXiv:2407.18418*.

Ming Xu. 2023. Medicalgpt: Training medical gpt model. https://github.com/shibing624/MedicalGPT.

Wenjing Yue, Xiaoling Wang, Wei Zhu, Ming Guan, Huanran Zheng, Pengfei Wang, Changzhi Sun, and Xin Ma. 2024. Tcmbench: A comprehensive benchmark for evaluating large language models in traditional chinese medicine. *Preprint*, arXiv:2406.01126.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2023. R-tuning: Teaching large language models to refuse unknown questions. *arXiv preprint arXiv:2311.09677*.

A Model Performance Analysis

A.1 Visualization of Model Performance

In this section, we present radar chart visualizations to highlight the performance of both closed-source and open-source models across different medication-related tasks. As shown in Figure 3 and Figure 4, the radar charts provide a clear, comparative view of how various models handle tasks such as Indication, Dosage and Administration, Contraindicated Population, and Mechanisms of Action. Notably, models such as GLM4 and XiaoYi stand out for their excellent performance, with XiaoYi leading the closed-source models and GLM4 demonstrating remarkable consistency. On the other hand, HuatuoGPT2 significantly outperforms the other open-source models. These findings underscore the importance of model selection in high-stakes domains like healthcare, where the quality of responses directly impacts patient safety.

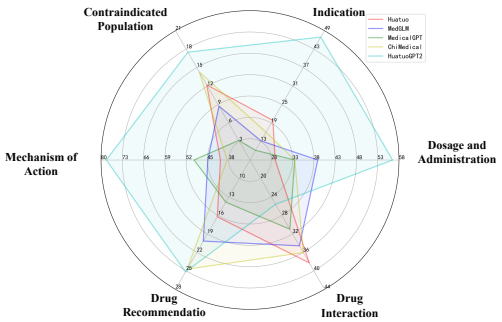


Figure 3: Radar Chart Representation of Open-Source Models Performance.

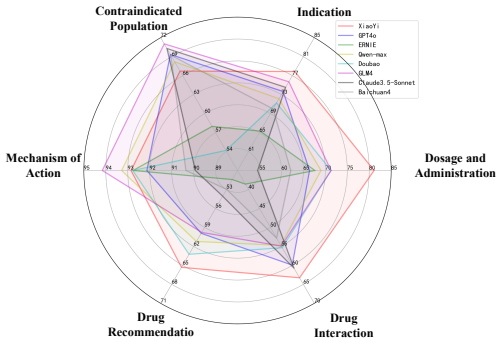


Figure 4: Radar Chart Representation of Close-Source Models Performance.

A.2 Knowledge Mastery Assessment of Common Drugs

To further evaluate model capabilities, we focus on a subset of 282 commonly used drugs. For each drug, we constructed questions about Indication, Dosage and Administration, Contraindicated Population, and Mechanism of Action, drawing from the benchmark dataset. The knowledge boundary of the models was then assessed by visualizing their performance on these tasks, as shown in the radar charts for GLM4, XiaoYi, and GPT4o, and the results are presented in Table 3.

Figure 5 illustrates the knowledge boundary performance of three models across 282 drugs, where each drug has 4 different sub-task questions. The radial score axis (0-4) represents the total number of correct answers per drug: Since each drug has 4 questions, a score of 4.0 means the model answered all 4 questions correctly.

Orange regions indicate that the model had only one chance to answer all 4 questions.

Yellow regions indicate that the model had up to 5 attempts per drug to answer correctly. If it answered any question correctly in any of these attempts, it was awarded 1 point.

The yellow regions that are not covered by orange represent cases where the model initially gave incorrect answers but later recovered and answered correctly within the 5 attempts. These visualizations emphasize that while some models show robustness in their knowledge, significant gaps remain in certain drug-related tasks.

The results indicate that GLM4 and XiaoYi exhibit stronger consistency in answering these questions correctly across the four tasks compared to GPT4o. However, there were cases where even the most advanced models struggled to demonstrate comprehensive knowledge across all aspects of these drugs. This highlights a key issue—despite their advanced capabilities, large models still fall short in areas of medication-related knowledge.

	Dosage and Administration	Indication	Contraindicated Population	Mechanisms of Action	Avg.
XiaoYi	**82.27**	**78.01**	63.12	92.20	**78.90**
GPT4o	67.02	74.11	70.2	92.91	76.24
ERNIE	69.15	64.89	58.51	91.13	70.92
Qwen-max	67.73	74.11	71.63	92.55	76.51
Doubao	75.89	74.47	64.54	92.55	76.86
GLM4	74.11	75.89	**71.99**	**93.97**	78.99
Claude3.5	67.09	67.73	54.26	93.46	70.64
Baichuan4	59.93	70.21	51.06	91.49	68.17

Table 3: Performance of various models on Common Drugs

A.3 Performance on Reasoning Models

While the GPT family is known for its strong reasoning capabilities, our results reveal nuanced performance differences, as shown in Table 4. In particular, OpenAI's o1 performs worse than GPT-4o, suggesting that strong reasoning ability alone does not guarantee superior performance, especially in knowledge-intensive domains like medicine. We argue that a model's performance ceiling in such domains is also closely tied to its parametric knowledge.

For instance, DeepSeek-R1 excels in tasks such as Drug Indication and Contraindicated Population, and this strength naturally extends to better performance in Medication Recommendation, which relies heavily on knowledge of drug usage constraints. In contrast, o1 performs worse in Indication, which correlates with its weaker performance in the Medication Recommendation task.

Furthermore, although GPT-4o is not explicitly trained on complex chain-of-thought (CoT) reasoning datasets, it demonstrates competitive reasoning ability. However, models like o1 and o3-mini, which are optimised for reasoning in code and mathematics, do not show a clear advantage in our medical reasoning benchmark, highlighting the limits imposed by insufficient medical knowledge.

To further validate that reasoning ability becomes the primary bottleneck when knowledge is sufficient, we constructed knowledge-complete prompts for Medication Recommendation and Drug Interaction tasks. These prompts explicitly included all necessary domain knowledge required to answer the questions.

As the results illustrated in Table 5 confirm, our hypothesis is supported: when provided with sufficient knowledge, models with strong reasoning capabilities, such as o1 and o3-mini, outperform GPT-4o, which lacks comparable reasoning-specific training. This highlights the importance of not only enriching models with knowledge but also enhancing their reasoning mechanisms—especially in professional domains like medicine.

In summary, our benchmark suggests that advancing domain-specific reasoning is a critical frontier for LLM development, and we hope our work offers meaningful insight for future research in this direction.

B Dataset Statistics

This section presents key statistics of our benchmark dataset across six task categories. As shown in Table 6, each task varies in terms of average input length, the number of unique drugs, and the number of associated diseases. Notably, the Drug Interaction task contains the longest samples and the largest set of drugs, reflecting its complexity. These statistics highlight the diversity and richness of our dataset, which is crucial for evaluating both the parametric knowledge and reasoning capabilities of LLMs in the medication domain.

C Expert Review of Datasets

During the manual verification phase, we hire 10 doctors with licensed qualifications to examine all the datasets we construct. We divided the 10 doctors into two groups, each consisting of five doctors. One group was responsible for reviewing the questions related to Drug Indication, Dosage and Administration, Contraindicated Population, and Mechanisms of Action, while the other group reviewed the questions for Medication Recommendation and Drug Interaction. After the initial review, the groups conducted a cross-check to ensure accuracy.

For data points that the doctors identified as problematic, we directly archived them and excluded

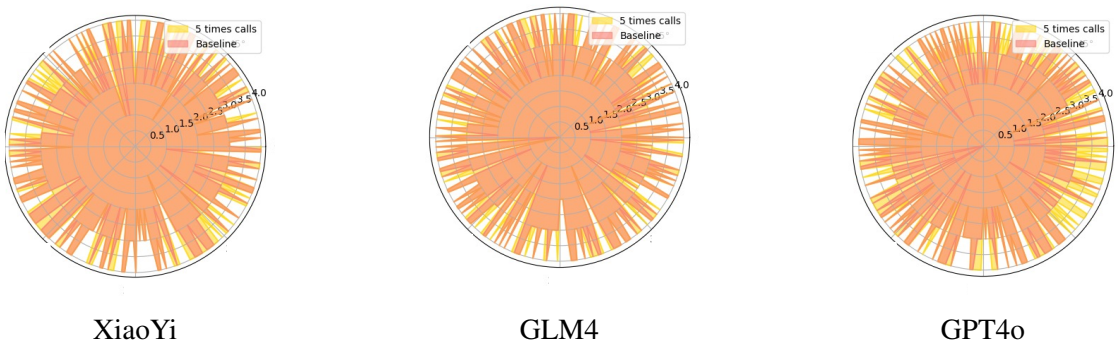


Figure 5: Knowledge boundary chart for GLM4, XiaoYi, and GPT4o across 282 common drugs. The orange area indicates that the model answered correctly once, while the yellow area indicates 5 times opportunities to answer correctly. The orange areas in the yellow-covered region represent cases where a model made an error in a single attempt but was able to recover after multiple tries.

Task	Dosage and Administration	Indication	Contraindicated Population	Mechanism of Action	Medication Recommendation	Drug Interaction	Average
DeepSeek-R1	70.81	77.06	75.69	91.98	61.59	62.75	73.31
OpenAI o1	56.69	74.80	58.31	92.93	61.53	45.10	64.89
OpenAI o3-mini	68.52	70.69	72.57	94.15	51.77	64.79	70.42
GPT-4o	66.41	73.65	69.35	92.13	59.79	59.93	70.21

Table 4: Zero-shot accuracy of reasoning models across medical knowledge tasks.

Model	**Medication Recommendation**	**Drug Interaction**
OpenAI o1	84.73	92.02
OpenAI o3-mini	80.79	91.34
GPT-4o	79.83	88.62

Table 5: Accuracy on reasoning tasks with knowledge-complete prompts.

Task	Average Length	Drugs Involved	Diseases Involved
Drug Indication	86.21	605	1061
Dosage and Administration	108.36	581	188
Contraindicated Population	94.45	607	636
Mechanisms of Action	108.25	791	224
Medication Recommendation	114.61	940	830
Drug Interaction	479.97	1293	None

Table 6: Task-wise statistics of the ChiDrug benchmark.

them from the final dataset. As a result, the accept rate of quesiton is 93.09%. The cost of hiring an doctor to label a single sample was 2 RMB (approximately 0.26 USD).

D Semantic Entropy (SE) Method for Knowledge Boundary Expression

In this section, we explore the Semantic Entropy (SE) method used to detect knowledge boundaries, as introduced in Section 4.2. The SE method is particularly noteworthy for its effectiveness in expressing model uncertainty and improving response reliability, as demonstrated in our experiments. We applied this method to HuatuoGPT2 and XiaoYi, observing that it significantly enhanced the models' performance on challenging tasks, such as Medication Recommendations and Drug Interactions.

As shown in Table 7, the SE method proved to be robust and consistent across different model architectures and sizes. It improved both Precision and Abstain Accuracy, regardless of the model's scale. This reinforces the notion that SE is an effective tool for managing uncertainty, making it an essential method for enhancing the reliability of models in real-world medical applications.

E Case Study

In Figure 6, we present a case to illustrate the practical section of the ChiDrug.

Model	Method	Dosage and Administration		Indication		Contraindicated Population		Mechanisms of Action		Medication Recommendation		Drug Interaction	
		Precision	A-Acc	Precision	A-Acc	Precision	A-Acc	Precision	A-Acc	Precision	A-Acc	Precision	A-Acc
HuatuoGPT2	w/o SE	55.83	55.83	47.03	47.03	18.66	18.66	77.16	77.16	25.18	25.18	25.60	25.60
	SE	64.79	55.32	70.28	52.71	28.11	26.95	86.24	83.09	29.22	24.95	38.56	31.72
XiaoYi	w/o SE	81.1	81.1	77.87	77.87	66.71	66.71	92.85	92.85	59.31	59.31	63.27	3.27
	SE	83.42	84.46	94.01	94.15	80.14	83.28	91.71	92.1	67.71	75.71	63.85	69.02

Table 7: Application of the SE method on HuatuoGPT2 and XiaoYi models, showcasing the performance improvements achieved through the SE method. This method enhances precision and uncertainty handling, effectively reducing hallucinations.

Tasks	Question	Answer
Dosage and Administration	复方锌布颗粒剂的推荐服用方式为多少包，多少次一天？ (A)儿童5岁以下一次2包，一天3次 (B)成人一次2包，一天3次 (C)成人一次1包，一天3次 (D)6～14岁儿童一次1包，一天2次 (E)6～14岁儿童一次1包，一天3次	B 用法用量： 口服。3～5岁儿童，一次半包；6～14岁儿童，一次1包；成人，一次2包，一日3次。
Indication	复方锌布颗粒剂主要用于缓解以下哪些症状？ (A)普通感冒引起的发热 (B)急性肠胃炎 (C)普通感冒引起的四肢酸痛 (D)普通感冒引起的打喷嚏	ACD 适应症： 用于缓解普通感冒或流行性感冒引起的发热、头痛、四肢酸痛、鼻塞、流涕、打喷嚏等症状。
Contraindicated Population	复方锌布颗粒剂不适用于以下哪些人群？ (A)心脏病患者 (B)哺乳期妇女 (C)对阿司匹林过敏的哮喘患者 (D)高血压患者	BC 禁忌： 1.对其他非甾体抗炎药过敏者禁用。2.孕妇及哺乳期妇女禁用。3.对阿司匹林过敏的哮喘患者禁用。
Mechanism of Action	关于复方锌布颗粒剂各组分的主要药理作用是： (A)布洛芬具有抗炎作用，葡萄糖酸锌促进蛋白质合成，马来酸氯苯那敏为解热镇痛药 (B)布洛芬具有解热镇痛作用，葡萄糖酸锌能增强免疫功能，马来酸氯苯那敏为抗组胺药 (C)布洛芬为抗组胺药，葡萄糖酸锌具有解热功能，马来酸氯苯那敏具有镇痛作用 (D)布洛芬为解热镇痛药，葡萄糖酸锌参与多种酶的合成与激活，马来酸氯苯那敏为抗组胺药	D 作用机制： 布洛芬能抑制前列腺素合成，具有解热镇痛作用；葡萄糖酸锌中锌离子能参与多种酶的合成与激活，有增强吞噬细胞的吞噬能力的作用；马来酸氯苯那敏为抗组胺药，能减轻由感冒或流感引起的鼻塞、流涕、打喷嚏等症状。
Drug Recommendation	回答以下不定项选择题（可能包含1个或多个正确选项）： 一位孕晚期患者因为感冒出现咳嗽、喉咙里有异物感以及扁桃体发炎，可以考虑推荐的药物是： (A)双黄连口服液 (B)热毒宁注射液 (C)贝美前列素滴眼液 (D)银芩胶囊	A 双黄连口服液的适应症： 疏风解表，清热解毒。用于外感风热所致的感冒，症见发热、咳嗽、咽痛。
Drug Interaction	注射用降纤酶与抗纤溶药联用的风险等级是？	高风险 注射用降纤酶 使用本品应避免与水杨酸类药物（如：阿司匹林）合用。抗凝血药可加强本品作用，引起意外出血；抗纤溶药可抵消本品作用，禁止联用。

Figure 6: Partial cases of ChiDrug on 6 sub datasets.

Rethinking Evaluation Metrics for Grammatical Error Correction: Why Use a Different Evaluation Process than Human?

Takumi Goto, Yusuke Sakai, Taro Watanabe
Nara Institute of Science and Technology
{goto.takumi.gv7, sakai.yusuke.sr9, taro}@is.naist.jp

Abstract

One of the goals of automatic evaluation metrics in grammatical error correction (GEC) is to rank GEC systems such that it matches human preferences. However, current automatic evaluations are based on procedures that diverge from human evaluation. Specifically, human evaluation derives rankings by aggregating sentence-level relative evaluation results, e.g., pairwise comparisons, using a rating algorithm, whereas automatic evaluation averages sentence-level absolute scores to obtain corpus-level scores, which are then sorted to determine rankings. In this study, we propose an aggregation method for existing automatic evaluation metrics which aligns with human evaluation methods to bridge this gap. We conducted experiments using various metrics, including edit-based metrics, n-gram based metrics, and sentence-level metrics, and show that resolving the gap improves results for the most of metrics on the SEEDA benchmark. We also found that even BERT-based metrics sometimes outperform the metrics of GPT-4. The proposed ranking method is intergrated GEC-METRICS[1].

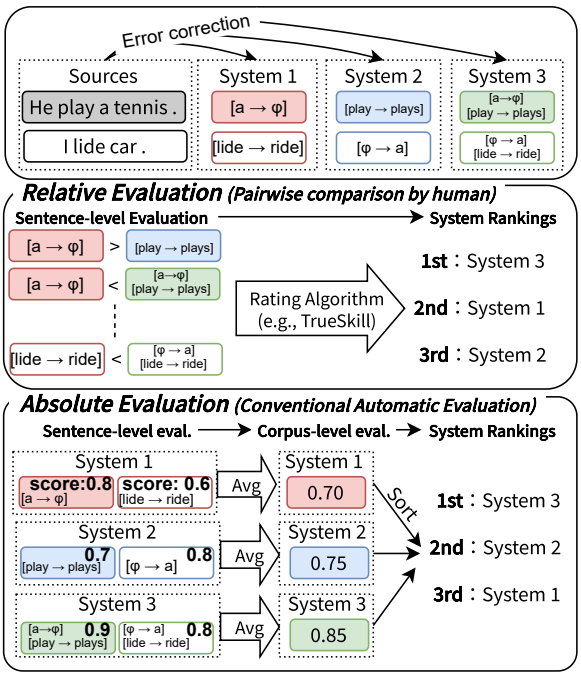


Figure 1: An overview of current human and automatic evaluation when ranking three GEC systems based on a dataset containing two sentences. Each system output represents edits for simplicity.

1 Introduction

Grammatical error correction (GEC) task aims to automatically correct grammatical errors and surface errors such as spelling and orthographic errors in text. Various GEC systems have been proposed based on sequence-to-sequence models (Katsumata and Komachi, 2020; Rothe et al., 2021), sequence tagging (Awasthi et al., 2019; Omelianchuk et al., 2020), and language models (Kaneko and Okazaki, 2023; Loem et al., 2023), and it is crucial to rank those systems based on automatic evaluation metrics to select the best system matching user's demands. Automatic evaluation is expected to rank GEC systems aligning with human preference, as evidenced by meta-evaluations of automatic metrics that assess their agreement with human evaluation (Grundkiewicz et al., 2015; Kobayashi et al., 2024b). For example, one can compute Spearman's rank correlation coefficient between the rankings produced by automatic and human evaluation, considering a metric with a higher correlation as a better metric.

However, despite the clear goal of reproducing human evaluation, current automatic evaluation is based on procedures that diverge from human evaluation. Figure 1 illustrates the evaluation procedure for ranking three GEC systems using a dataset comprising two sentences. In human evaluation, corrected sentences generated for the same input sentence are compared relatively across sys-

[1]A library for GEC evaluation proposed by Goto et al. (2025), https://github.com/gotutiyan/gec-metrics.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1165–1172
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

tem outputs, i.e., pairwise comparison, and the results are aggregated as rankings using rating algorithms such as TrueSkill (Herbrich et al., 2006). In contrast, automatic evaluation estimates sentence-wise scores, then averages them at the corpus level and determines rankings by sorting these averaged scores. As such, current automatic evaluation follows a procedure that deviates from human evaluation, contradicting the goal of reproducing human judgment. Intuitively, it would be desirable for automatic evaluation to follow the same procedure as human evaluation.

In this study, we hypothesize that resolving this gap will more closely align automatic evaluation with human evaluation. Based on this hypothesis, we propose computing rankings in automatic evaluation using the same procedure as human evaluation, e.g., using TrueSkill after deriving pairwise estimates based on sentence-wise scores when human evaluation employs TrueSkill. In our experiments, we conducted a meta-evaluation on various existing automatic evaluation metrics using the SEEDA dataset (Kobayashi et al., 2024b), that is a representative meta-evaluation benchmark. The results show that bridging the identified gap improves ranking capability for many metrics and that BERT-based (Devlin et al., 2019) automatic evaluation metrics can even outperform large language models (LLMs), GPT-4 (OpenAI et al., 2024), in evaluation. Furthermore, we discuss the use and development of automatic evaluation metrics in the future, emphasizing that sentence-level relative evaluation is particularly important for developing new evaluation metrics.

2 Gap Between Human and Automatic Evaluation

2.1 Background

Human evaluation has been conducted by Grundkiewicz et al. (2015), who manually evaluated systems submitted to the CoNLL-2014 shared task (Ng et al., 2014), and by Kobayashi et al. (2024b), who included state-of-the-art GEC systems such as LLMs in their dataset. In both studies, system rankings were derived by applying a rating algorithm to sentence-level pairwise comparisons. Commonly used rating algorithms include Expected Wins (Bojar et al., 2013) and TrueSkill (Herbrich et al., 2006; Sakaguchi et al., 2014). Grundkiewicz et al. (2015) adopted Expected Wins as their final ranking method, whereas Kobayashi et al. (2024b) used TrueSkill to determine the final ranking. Kobayashi et al. (2024b) also pointed out the importance of aligning the granularity of evaluation between automatic evaluation and human evaluation, but did not mention the procedure for converting sentence-level evaluation into system rankings.

Automatic evaluation is conducted using various evaluation metrics, including reference-based and reference-free approaches, as well as sentence-level and edit-based metrics. Most of these metrics follow a procedure in which each sentence is assigned an absolute score, which is then aggregated into a corpus-level evaluation score. For example, sentence-level metrics such as SOME (Yoshimura et al., 2020) and IMPARA (Maeda et al., 2022) aggregate scores by averaging, while edit-based metrics such as ERRANT (Felice et al., 2016; Bryant et al., 2017) and GoToScorer (Gotou et al., 2020), as well as n-gram-based metrics such as GLEU (Napoles et al., 2015, 2016) and GREEN (Koyama et al., 2024), aggregate scores by accumulating the number of edits or n-grams. The corpus-level scores obtained through these methods can be converted into system rankings by sorting.

2.2 How to Resolve the Gap?

The gap can be simply addressed by using automatic evaluation metrics in the same manner as human evaluation. Given that the SEEDA dataset uses TrueSkill as the aggregation method, we will close the gap by using TrueSkill for automated evaluation as well. First, since existing automatic evaluation metrics compute sentence-wise scores, we convert these scores into pairwise comparison results. For example, in the case illustrated in Figure 1, the evaluation scores of 0.8, 0.7, and 0.9 for corrected sentences corresponding to the first sentence ("He play a tennis") can be compared to produce pairwise comparison results similar to those in human evaluation. Next, we compute system rankings by applying TrueSkill to the transformed pairwise comparison results. In this study, we consider all combinations of pairwise comparisons for system set. That is, given N systems, a total of $N(N-1)$ comparisons are performed per sentence, and system rankings are computed based on these results including ties.

A similar method was employed by Kobayashi et al. (2024a), but they did not mention the gap. Also, their experiments used the TrueSkill aggregation for their proposed LLM-based metrics, but used conventional aggregation methods, e.g., aver-

aging, for other metrics. We discuss and organize the gap between human and automatic evaluation in detail, and then solve the gap by applying TrueSkill to all metrics for fair comparison.

Note that our method is explained using TrueSkill, which is used as the human evaluation method for SEEDA. If another meta-evaluation dataset uses a different aggregation method, such as Expected Wins, we should use that instead. We emphasize that our claim is the importance of aligning the aggregation methods of human and automatic evaluations, as even this simple practice has been largely overlooked so far.

3 Experiments

3.1 Automatic Evaluation Metrics

We provide more detailed experimental settings for each metric in Appendix A.

Edit-based metrics We use ERRANT (Felice et al., 2016; Bryant et al., 2017) and PT-ERRANT (Gong et al., 2022). Both are reference-based evaluation metrics that assess at the edit level. When multiple references are available, the reference that yields the highest $F_{0.5}$ score is selected for each sentence.

n-gram based metrics We use GLEU+ (Napoles et al., 2015, 2016) and GREEN (Koyama et al., 2024). The n-gram overlap is checked among the input sentence, hypothesis sentence, and reference sentence. When multiple references are available, GLEU+ uses the average score across all references, and GREEN uses the reference that yields the highest score is selected for each sentence.

Sentence-level metrics SOME (Yoshimura et al., 2020), IMPARA (Maeda et al., 2022), and Scribendi Score (Islam and Magnani, 2021) are used. All of them are based on small neural models such as $\text{BERT}_{\text{base}}$ (Devlin et al., 2019) and designed as a reference-free metric that considers the correction quality estimation score as well as the meaning preservation score between the input and corrected sentences.

3.2 Meta-Evaluation Method

We use the SEEDA dataset (Kobayashi et al., 2024b) for meta-evaluation. Meta-evaluation results are reported based on human evaluation results using TrueSkill for both the sentence-level human-evaluation, SEEDA-S, and the edit-level human-evaluation, SEEDA-E. Additionally, we also report results for both the `Base` configuration, which excludes the fluent reference and GPT-3.5 outputs that allow for larger rewrites, and the `+Fluency` configuration, which includes them.

Furthermore, we evaluate the robustness of the calculated rankings using window analysis (Kobayashi et al., 2024a). The window analysis computes correlation coefficients only for consecutive N systems, after sorting systems based on human evaluation results. This allows us to analyze whether automatic evaluation can correctly assess a set of systems that appear to have similar performance from the human evaluation. In this study, we perform it with $N = 8$ for 14 systems corresponding to the `+Fluency` configuration, and report both Pearson and Spearman correlation coefficients. That is, correlation coefficients are computed for the rankings 1 to 8, 2 to 9, . . ., and 7 to 14 from human evaluation.

3.3 Experimental Results

Table 1 shows the results of the meta-evaluation. The upper group presents evaluation results based on the conventional method of averaging or summing, and the bottom group presents results evaluated using TrueSkill, which follows the same evaluation method as human evaluation. The bottom group includes the evaluation results based on GPT-4 reported by Kobayashi et al. (2024a), which correspond to the state-of-the-art metrics.

The overall trend indicates that using TrueSkill-based evaluation improves the correlation coefficients for most of metrics. In particular, the results of IMPARA in the SEEDA-S and `+Fluency` setting outperformed those of GPT-4 results. Additionally, ERRANT showed an improvement of more than 0.2 points in many configurations. These results show that using automatic evaluation metrics with the same evaluation procedure as human evaluation makes the ranking closer to human evaluation. In other words, the existing automatic evaluation metrics were underestimated in the prior reports due to the gap in the meta-evaluation procedure.

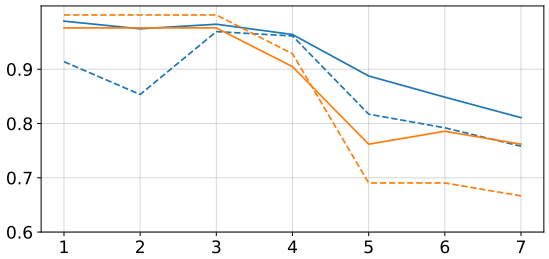
Unlike edit-level or sentence-level metrics, no effect was observed in n-gram-level metrics such as GLEU+ and GREEN. This stems from the poor sentence-level evaluation performance of n-gram based metrics. The BLEU paper (Papineni et al., 2002), which is a n-gram-level metric for machine translation and the basis of the GLEU+, notes that the brevity penalty can excessively penalize scores for short sentences, and uses corpus-level lengths

Metrics	SEEDA-S				SEEDA-E			
	Base		+Fluency		Base		+Fluency	
	r (Pearson)	ρ (Spearman)	r	ρ	r	ρ	r	ρ
w/o TrueSkill								
ERRANT	0.545	0.343	-0.591	-0.156	0.689	0.643	-0.507	0.033
PTERRANT	0.700	0.629	-0.546	0.077	0.788	0.874	-0.470	0.231
GLEU+	0.886	0.902	0.155	0.543	0.912	0.944	0.232	0.569
GREEN	0.925	0.881	0.185	0.569	0.932	0.965	0.252	0.618
SOME	0.892	0.867	0.931	0.916	0.901	0.951	0.943	0.969
IMPARA	0.916	0.902	0.887	0.938	0.902	0.965	0.900	0.978
Scribendi	0.620	0.636	0.604	0.714	0.825	0.839	0.715	0.842
w/ TrueSkill								
ERRANT	0.763	0.706	-0.463	0.095	0.881	0.895	-0.374	0.231
PTERRANT	0.870	0.797	-0.366	0.182	0.924	0.951	-0.288	0.279
GLEU+	0.863	0.846	0.017	0.393	0.909	0.965	0.102	0.486
GREEN	0.855	0.846	-0.214	0.327	0.912	0.965	-0.135	0.420
SOME	0.932	0.881	0.971	0.925	0.893	0.944	0.965	0.965
IMPARA	0.939	**0.923**	**0.975**	**0.952**	0.901	0.944	0.969	0.965
Scribendi	0.674	0.762	0.745	0.859	0.837	0.888	0.826	0.912
GPT-4-E (fluency)	0.844	0.860	0.793	0.908	0.905	**0.986**	0.848	0.987
GPT-4-S (fluency)	0.913	0.874	0.952	0.916	**0.974**	0.979	**0.981**	**0.982**
GPT-4-S (meaning)	**0.958**	0.881	0.952	0.925	0.911	0.960	0.976	0.974

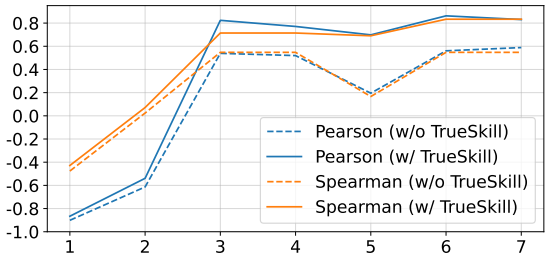
Table 1: Correlation with human evaluation using the SEEDA dataset. *w/o TrueSkill* refers to the conventional evaluation procedure, while *w/ TrueSkill* represents the proposed evaluation procedure. Improvements over the conventional procedure are underlined, and the highest value in each column is highlighted in **bold**. The GPT-4 results refer to those reported in Kobayashi et al. (2024b).

to address this issue. Since GLEU+ also employs a brevity penalty, it cannot accurately calculate sentence-level scores depending on sentence length. This is a serious issue for TrueSkill-based aggregation because the quality of sentence-level scores directly affects the quality of system rankings. Furthermore, while GREEN does not use a brevity penalty, the score can become unstable as the "n" for n-gram increases, especially for short sentences. This has a negative impact on the geometric mean among n-gram scores, which is the final score. An ideal metric should provide evaluation results that better align with human judgments when ranking systems in the same way humans do. Given this premise, our results suggest a human alignment issue of n-gram-level metrics.

Figure 2 shows the results of the window analysis for IMPARA and ERRANT measured on SEEDA-S and SEEDA-E, respectively. From Figure 2a, it can be seen that IMPARA particularly aligns with human evaluation in the lower ranks. The Pearson correlation coefficient also showed an improvement in the evaluation results for the top systems as well. Since the top systems include GEC systems that are largely rewritten, such as GPT-3.5, this characteristic is useful, considering that LLM-based correction methods will become popular in the future. Figure 2b shows that ERRANT consistently showed improved correlation coefficients with the proposed method, but still struggled with evaluating the top systems. For edit-based evaluation metrics, it is still considered dif-



(a) IMPARA



(b) ERRANT

Figure 2: The results of the window analysis for $N = 8$ are shown. The x-axis represents the starting rank of human evaluation. For example, $x = 2$ shows the results for the systems ranked 2nd to 10th in human evaluation.

ficult to assess such GEC systems even with the evaluation method aligned with human evaluation[2].

4 Conclusion

In this study, we focused on the fact that human evaluation aggregates sentence-level scores into system rankings based on TrueSkill, while automatic evaluation uses a different evaluation, and we proposed to use TrueSkill in automatic evaluation as well. Experimental results with various existing metrics showed improvements in correlations with human evaluation for many of the metrics, indicating that agreement on the aggregation method is important. Our core statement in this paper is not about using TrueSkill, but rather the importance of using the same aggregation method as human evaluation. For instance, if future meta-evaluation datasets switch the aggregation method for human evaluation to averaging sentence-level scores, then automatic evaluation should likewise adopt the same approach.

Given the discussion so far, at least in the current situation of GEC evaluation, we recommend transitioning the aggregation method from averaging or summing to using a rating algorithm, such as TrueSkill. We also recommend that evaluation metrics should be developed that allow for accurate sentence-wise comparisons. This is evidenced by the fact that IMAPARA achieves a higher correlation coefficient than SOME in Table 1. In fact, IMAPARA is trained to assess the pairwise comparison results, whereas SOME is trained to evaluate sentences absolutely.

Limitations

Use for Purposes Other Than System Ranking The proposed method is designed for system ranking and cannot be used for other types of evaluation, such as analyzing the strengths and weaknesses of a specific system. For instance, when analyzing whether a model excels in precision or recall, it is more useful to accumulate the number of edits at the corpus level, as done in existing evaluation methods.

Reproducing the Outputs of Compared GEC Systems Since the proposed ranking method requires inputting all GEC outputs being compared, it is necessary to reproduce their models. This point is different from existing absolute evaluation methods, where previously reported scores can be cited. While this may seem burdensome for researchers, it can also be seen as an important step toward promoting the publication of reproducible research results.

[2]Using a larger number of references may solve this issue.

Ethical Considerations

When the metric contains social biases, the proposed method cannot eliminate that bias and may reflect that bias in the rankings. However, we argue that this problem should be resolved as a metric problem.

Acknowledgments

We sincerely thank the anonymous reviewers for their insightful comments and suggestions. This work has been supported by JST SPRING. Grant Number JPMJSP2140.

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.

Peiyuan Gong, Xuebo Liu, Heyan Huang, and Min Zhang. 2022. Revisiting grammatical error correction evaluation and beyond. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6891–6902, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025. gec-metrics: A unified library for grammatical error correction evaluation. *Preprint*, arXiv:2505.19388.

Takumi Gotou, Ryo Nagata, Masato Mita, and Kazuaki Hanawa. 2020. Taking the correction difficulty into account in grammatical error correction evaluation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2085–2095, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Edward Gillian. 2015. Human evaluation of grammatical error correction systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 461–470, Lisbon, Portugal. Association for Computational Linguistics.

Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. Trueskill™: A bayesian skill rating system. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

Md Asadul Islam and Enrico Magnani. 2021. Is this the end of the gold standard? a straightforward referenceless grammatical error correction metric. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3009–3015, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Masahiro Kaneko and Naoaki Okazaki. 2023. Reducing sequence length by predicting edit spans with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10017–10029, Singapore. Association for Computational Linguistics.

Satoru Katsumata and Mamoru Komachi. 2020. Stronger baselines for grammatical error correction using a pretrained encoder-decoder model. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 827–832, Suzhou, China. Association for Computational Linguistics.

Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024a. Large language models are state-of-the-art evaluator for grammatical error correction. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 68–77, Mexico City, Mexico. Association for Computational Linguistics.

Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024b. Revisiting meta-evaluation for grammatical error correction. *Transactions of the Association for Computational Linguistics*, 12:837–855.

Shota Koyama, Ryo Nagata, Hiroya Takamura, and Naoaki Okazaki. 2024. n-gram F-score for evaluating grammatical error correction. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 303–313, Tokyo, Japan. Association for Computational Linguistics.

Mengsay Loem, Masahiro Kaneko, Sho Takase, and Naoaki Okazaki. 2023. Exploring effectiveness of GPT-3 in grammatical error correction: A study on performance and controllability in prompt-based methods. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 205–219, Toronto, Canada. Association for Computational Linguistics.

Koki Maeda, Masahiro Kaneko, and Naoaki Okazaki. 2022. IMPARA: Impact-based metric for GEC using parallel data. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3578–3588, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593, Beijing, China. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2016. Gleu without tuning. *Preprint*, arXiv:1605.02592.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 1–11, Baltimore, Maryland, USA. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. SOME: Reference-less sub-metrics optimized for manual evaluations of grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6516–6522, Barcelona, Spain (Online). International Committee on Computational Linguistics.

A Detailed Experimental Settings for Evaluation Metrics

We used GEC-METRICS (Goto et al., 2025) for the implementation. The detailed experimental settings are as follows:

ERRANT Evaluations were conducted using the Python module `errant==3.0.0` with the span-based correction setting. Although the edits in the CoNLL-2014 references were manually annotated, they were re-extracted using ERRANT. This is important for an evaluation based on consistent edits.

PT-ERRANT We employed the $F1$ score of BERTScore for the weighting. The baseline rescaling was adopted, and IDF adjustment was not performed. We used `bert-base-uncased` for the BERT model. These settings are consistent with those of the official implementation [3]. Similar to ERRANT, we re-extracted reference edits via `errant` module.

GLEU+ We used **GLEU+** with n-grams up to 4-grams and 500 iterations during reference sampling.

GREEN Similarly, we used **GREEN** with n-grams up to 4-grams, utilizing the $F_{2.0}$ score, following those employed by Koyama et al. (2024).

SOME We used the official models for grammaticality, fluency, and meaning preservation, with respective weights of 0.55, 0.43, and 0.02. These weights correspond to those tuned by Yoshimura et al. (2020) for sentence-level evaluation performance.

IMAPARA As a pre-trained quality estimation model was not publicly available, we newly constructed it through reimplementation and experimentation. Following Maeda et al. (2022), the CoNLL-2013 dataset (Ng et al., 2013) was used as a seed corpus and was split into training, development, and evaluation sets with an 8:1:1 ratio. We fine-tuned `bert-base-cased`, and followed `BertForSequenceClassification` from the Transformers library [4] (Wolf et al., 2020) for the classifier architecture. This corresponds to transforming the `CLS` representation of the BERT model into a real value via a single projection layer. During inference, `bert-base-cased` was

[3] https://github.com/pygongnlp/PT-M2

[4] https://github.com/huggingface/transformers

used for the similarity estimation model, with a threshold set to 0.9.

Scribendi Score GPT-2 (Radford et al., 2019)[5] was utilized as the language model, and the threshold for the maximum of the Levenshtein-distance ratio and token sort ratio was set to 0.8.

[5] https://huggingface.co/openai-community/gpt2

Learning Auxiliary Tasks Improves Reference-Free Hallucination Detection in Open-Domain Long-Form Generation

Chengwei Qin[†♦*], **Wenxuan Zhou**[♣], **Karthik Abinav Sankararaman**[♣], **Nanshu Wang**[♣], **Tengyu Xu**[♣], **Alexander Radovic**[♣], **Eryk Helenowski**[♣], **Arya Talebzadeh**[♣], **Aditya Tayade**[♣], **Sinong Wang**[♣], **Shafiq Joty**[♦], **Han Fang**[♣], **Hao Ma**[♣]

[†]The Hong Kong University of Science and Technology (Guangzhou)

[♦]Nanyang Technological University [♣]GenAI, Meta

Abstract

Hallucination, the generation of factually incorrect information, remains a significant challenge for large language models (LLMs), especially in open-domain long-form generation. Existing approaches for detecting hallucination in long-form tasks either focus on limited domains or rely heavily on external fact-checking tools, which may not always be available.

In this work, we systematically investigate reference-free hallucination detection in open-domain long-form responses. Our findings reveal that internal states (*e.g.,* model's output probability and entropy) alone are insufficient for reliably (*i.e.,* better than random guessing) distinguishing between factual and hallucinated content. To enhance detection, we explore various existing approaches, including prompting-based methods, probing, and fine-tuning, with fine-tuning proving the most effective. To further improve the accuracy, we introduce a new paradigm, named RATE-FT, that augments fine-tuning with an auxiliary task for the model to jointly learn with the main task of hallucination detection. With extensive experiments and analysis using a variety of model families & datasets, we demonstrate the effectiveness and generalizability of our method, *e.g.,* +3% over general fine-tuning methods on LongFact.

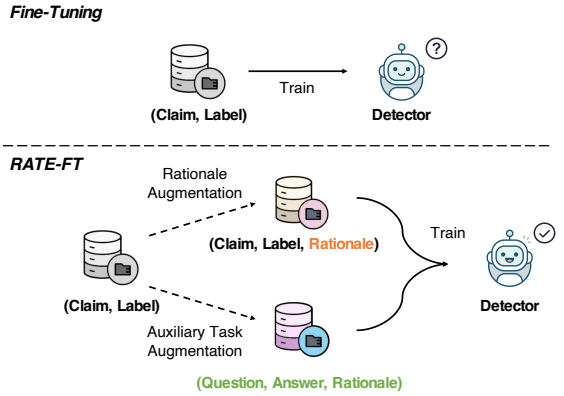


Figure 1: Comparison between Fine-Tuning and RATE-FT for hallucination detection. RATE-FT improves Fine-Tuning by incorporating rationales and an auxiliary task (question answering) into the training process.

1 Introduction

With the recent advancements in model scale and pretraining data, large language models (LLMs) have demonstrated remarkable capabilities in various natural language processing (NLP) tasks (Brown et al., 2020). Despite these successes, hallucination, where models tend to produce content that conflicts with real-world facts, remains a significant challenge (Zhang et al., 2023). Most existing research on hallucination detection has focused on short-form tasks, where the output consists of one or a few tokens. While these methods are effective for short-form content (Manakul et al., 2023; Mahaut et al., 2024; Yehuda et al., 2024; Zhang et al., 2024a), extending them to open-domain long-form generation presents additional complexities and new challenges. Unlike short-form tasks, long-form responses can span hundreds or even thousands of tokens, requiring models to generate detailed and nuanced answers to broad fact-seeking prompts (Wei et al., 2024). This necessitates that LLMs synthesize information across multiple knowledge domains, increasing the risk of generating content that sounds plausible yet is factually incorrect. For example, when answering 'What is the significance of Amber Room?', LLMs may generate responses that mix accurate historical information with fabricated details, complicating the task of distinguishing fact from hallucination.

Recent efforts have sought to address hallucination detection in long-form tasks. However, they either focus on limited domains, *e.g.,* biography generation (Min et al., 2023; Fadeeva et al., 2024) or rely heavily on external fact-checking tools or

[*]Correspondence to Chengwei Qin <qcwthu@gmail.com> and Karthik Abinav Sankararaman <karthikabinavs@gmail.com>

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1173–1182
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

knowledge bases, *e.g.,* Google Search (Wei et al., 2024). While these tools offer valuable support, they are not always available or scalable. This raises an important question: *can we develop hallucination detectors that rely solely on the model itself, without the need for external fact-checking resources?* So far, little attention has been given to systematically exploring how the model's own mechanisms can be used for detecting hallucinations in open-domain long-form generation.

To address this gap, we start by investigating hallucination detection in open-domain long-form responses using the model's internal states, *e.g.,* output probability and entropy. Specifically, we decompose long-form responses into atomized claims using the model and verify each claim's correctness using Google Search to construct benchmark data following Wei et al. (2024). Our analysis reveals that these internal states alone are insufficient for reliably (*i.e.,* better than random guessing) distinguishing between correct and incorrect claims, indicating that the mechanisms for detecting hallucinations in long-form outputs differ significantly from those in short-form tasks. To enhance detection, we explore several existing methods, including prompting, probing, and fine-tuning LLMs. Our experimental results show that fine-tuning LLMs is the most effective method to detect hallucinations.

Building on this, we introduce a novel method Rationale and Auxiliary Task Enhanced Fine-Tuning (RATE-FT) (Figure 1). Specifically, we convert the original claims into auxiliary question answering (QA) examples for augmentation, providing a complementary learning perspective for the model, which enables better generalization. Additionally, we incorporate collected rationales into the training process for better reasoning. Extensive experiments and analysis using different models demonstrate the effectiveness and generalizability of our approach. Furthermore, we investigate the integration of model uncertainty into hallucination detection in Appendix A.8. In summary, our main contributions are:

- We are the first to systematically investigate reference-free hallucination detection in open-domain long-form generation by analyzing a representative set of existing methods.
- We introduce a novel approach that incorporates rationales and an auxiliary question answering task into fine-tuning, achieving significant performance improvements.

2 Related Work

Large Language Models (LLMs) often generate content that appears plausible but is factually unsupported, a phenomenon commonly known as hallucination (Zhang et al., 2023). Based on whether the hallucinated content contradicts real-world facts or the input context, hallucination can be categorized into two main groups: factuality hallucination and faithfulness hallucination (Huang et al., 2023). Extensive research has been conducted on exploring the causes (Onoe et al., 2022; Kang and Choi, 2023; Wei et al., 2023; Liu et al., 2024), detection (Min et al., 2023; Zhao et al., 2023; Chen et al., 2024a; Fadeeva et al., 2024; Wei et al., 2024), and mitigation (Gao et al., 2023; Ji et al., 2023; Tian et al., 2024; Zhang et al., 2024b; Kang et al., 2024; Lin et al., 2024) of hallucination in LLMs. However, most existing hallucination detection methods have primarily focused on short-form tasks, where the output consists of one or a few tokens. In this work, we shift the focus to the more challenging problem of reference-free hallucination detection in open-domain long-form generation, where outputs are substantially longer and require a more nuanced evaluation of actuality.

3 Are LLMs' Internal States Sufficient for Open-Domain Long-Form Generation?

The internal states of LLMs, such as output probability and entropy, have been shown to be effective in detecting hallucinations in short-form tasks, where outputs are typically limited to only a few tokens. By analyzing these signals, models can often differentiate between factual and hallucinated information. However, their applicability in open-domain long-form generation remains underexplored. A key question is whether LLMs can depend solely on their internal states to identify hallucinations in long-form generation, without using external fact-checking tools. To answer it, we conduct some pilot experiments on LongFact (Wei et al., 2024), a long-form generation dataset spanning 38 different domains. Specifically, for each prompt in the sampled subset (200 prompts), we obtain a long-form response from Llama-3-8B-Instruct with greedy decoding. Following Wei et al. (2024), we employ the model to decompose long-form responses into atomized claims and label them as 'factual' or 'hallucinated' together with the reasons (see Appendix A.1 for construction details).

For each claim, we mainly focus on two types

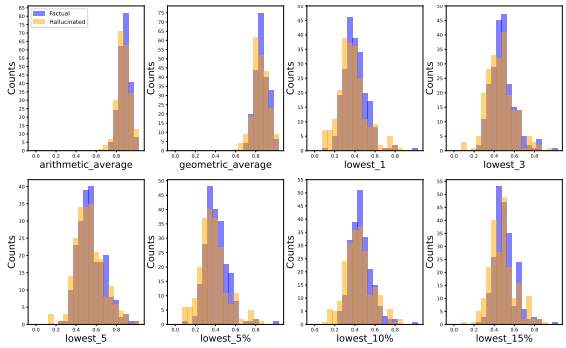


Figure 2: Detection results based on token probability.

of internal states to estimate factual confidence following SelfCheckGPT (Manakul et al., 2023): the probability or the entropy (uncertainty) of output tokens. Specifically, we examine the arithmetic and geometric [1] averages of all tokens, the average of tokens with the top-K lowest probability or highest entropy ($K = 1, 3, 5$), and the average of tokens with the top-$P\%$ lowest probability or highest entropy ($P = 5, 10, 15$). The results in Figure 2 and Appendix A.2 suggest that neither internal state reliably, *i.e.,* better than random guessing, predicts the correctness of a given claim, which may be due to the presence of numerous insignificant tokens within the claim, such as stop words. To address this, we consider variants that focus only on output tokens related to entities. The results, shown in Appendix A.2, reveal similar patterns (see Appendix A.3 for a detailed comparison with the findings in Manakul et al. (2023)). We analyze the underlying reasons as follows. In open-domain long-form generation, claims are not limited to a few tokens, which introduces multiple sources of uncertainty. Specifically, *the probability or entropy reflects the model's confidence in how a claim is expressed, i.e., its confidence in the claim as a sequence of output tokens, rather than in the correctness of the claim.* Different surface forms of the claim yield different confidence levels, leading to unreliable estimates.

Considering the unreliability of LLMs' internal states in hallucination detection, there are several promising alternative approaches, including prompting, probing and fine-tuning LLMs, which we explore in the next section.

4 Prompting, Probing and Fine-Tuning

Based on a review of the research area, we identify three groups of existing hallucination detection methods, which we discuss below.

Prompting Prompting-based approaches involve directly prompting LLMs to assess the correctness of a given claim without additional training. We investigate the following three different methods: (*i*) Prompting the model to output *'True' or 'False'* for a given claim, referred to as $\text{Prompt}_{\text{TF}}$. The probability assigned to the token 'True' represents P_{factual}, while the probability assigned to 'False' represents $P_{\text{hallucinated}}$. (*ii*) Prompting the model to output the *probability* that it considers the given claim to be correct, referred to as $\text{Prompt}_{\text{Prob}}$. This number directly represents P_{factual}. (*iii*) SelfCheck-GPT, which detects hallucinations by sampling additional responses from the model and assessing inconsistencies between each response and the target claim. The *proportion* of responses that support the claim is taken as P_{factual}. Following Manakul et al. (2023), we sample 20 responses for detection.

Probing Following Su et al. (2024a), we train a multilayer perceptron (MLP) on the contextualized embeddings of LLMs to perform binary classification for hallucination detection, while keeping the base LLM frozen. The trained MLP outputs P_{factual} as an indicator for classification.

Fine-Tuning We fine-tune the base LLM with LoRA to enhance its ability to output 'True' or 'False' for a given claim (Kapoor et al., 2024). Similar to $\text{Prompt}_{\text{TF}}$, the probabilities assigned to the tokens 'True' and 'False' correspond to P_{factual} and $P_{\text{hallucinated}}$, respectively. Note that LoRA fine-tuning allows us to easily use the original model for general tasks while applying the trained LoRA specifically for hallucination detection.

Following the data construction process outlined in Appendix A.1, we conduct experiments on the full set of LongFact using Llama-3-8B-Instruct. This process yields 2,711 factual and hallucinated claims, which are subsequently split into training (70%), validation (20%), and test (10%) sets. For all three types of methods, we use P_{factual} as the classification indicator. Specifically, a claim is classified as 'factual' if P_{factual} exceeds a predefined threshold; otherwise, it is classified as 'hallucinated'. The optimal threshold is determined through a search on the validation set. Consistent with Tang et al. (2024); Chen et al. (2024b), we employ balanced accuracy (BAcc) as the evaluation metric: $\text{BAcc} = \frac{1}{2}(\frac{\text{TP}}{\text{TP+FN}} + \frac{\text{TN}}{\text{TN+FP}})$, where TP, TN, FP, and FN stand for true/false positives/negatives.

The results of different methods on the test set, as

[1] Commonly known as perplexity

Dataset	Method				
	$\text{Prompt}_{\text{TF}}$	$\text{Prompt}_{\text{Prob}}$	SelfCheckGPT	Probing	Fine-Tuning
LongFact	69.9	53.4	69.1	74.4	**76.1**
Biography	72.3	56.3	71.9	77.0	**78.2**

Table 1: BAcc (%) of existing hallucination detection methods on LongFact and biography generation.

Dataset	Method				
	$\text{Prompt}_{\text{TF}}$	$\text{Prompt}_{\text{CoT-TF}}$	Probing	Fine-Tuning	RATE-FT
LongFact	69.9	74.9	74.4	76.1	**79.6**
Biography	72.3	74.8	77.0	78.2	**80.9**

Table 2: BAcc (%) of RATE-FT and baseline methods.

shown in Table 1, indicate that fine-tuning LLMs is the most effective among all existing methods (see Appendix A.4 for an analysis of fine-tuning effectiveness in Out-of-Distribution (OOD) scenarios). While both $\text{Prompt}_{\text{TF}}$ and SelfCheckGPT achieve decent performance, Probing yields notable improvements by incorporating additional training with labels obtained from external search. Fine-Tuning further enhances performance by updating the internal features of LLMs, enabling more effective learning. In contrast, $\text{Prompt}_{\text{Prob}}$ performs significantly worse, likely due to LLMs' tendency to output high probabilities for hallucinated claims, leading to overconfidence. Additionally, we extend the experiments to biography generation (Min et al., 2023). The results presented in Table 1 demonstrate that the observations and conclusions can be generalized to different datasets.

Building on these findings, a natural question arises: can Fine-Tuning be further improved to develop more effective hallucination detectors? We answer this question by *incorporating rationales and an auxiliary task into the training process.*

5 Rationale and Auxiliary Task Enhanced Fine-Tuning (RATE-FT)

While hallucination detection is not regarded as a reasoning task in the conventional sense, incorporating Chain-of-Thought (CoT) (Wei et al., 2022) explaining the judgment can still be beneficial for distinguishing factual content from hallucinated information as it enables LLMs to better evaluate the correctness of claims by systematically analyzing underlying components. To examine the impact of rationales, we prompt the model to generate a reasoning path before making a judgment (*i.e.,* 'True' or 'False'), referred to as $\text{Prompt}_{\text{CoT-TF}}$. This approach improves performance from 69.9 (using $\text{Prompt}_{\text{TF}}$) to 74.9, highlighting the effectiveness of incorporating CoT reasoning.

Augmenting Fine-Tuning with Rationales Building on the above observation, we augment the fine-tuning dataset with rationales generated by the model during data construction, explaining whether the search results support the claims. Notably, we adopt the 'label-rationale' format to maintain the same inference cost as the baseline Fine-Tuning. This allows us to directly derive P_{factual} from the first output token without requiring the generation of the complete reasoning path.

Consolidating knowledge through repetition in diverse contexts is a fundamental principle of effective human learning (Ausubel, 2012). For example, medical students deepen their understanding of anatomy by studying diagrams, practicing in simulations, and engaging in hands-on dissections, each offering a unique perspective on the same foundational knowledge. Drawing inspiration from this paradigm, we introduce an auxiliary question answering (QA) task into the fine-tuning process to further strengthen the model's understanding and enhance its generalization capabilities. This auxiliary QA task serves as a complementary component to the primary hallucination detection task, offering the model an alternative but closely related perspective on the problem (see Appendix A.5 for more analysis on the auxiliary task).

Augmenting Fine-Tuning with QA Task Specifically, for each claim, we first prompt the model to generate a question about the key information within it. If the claim is factual, we ask the model to extract the correct answer directly from the claim and provide an explanation, forming a QA example. For hallucinated claims, we leverage the augmented rationale to guide the model in generating an appropriate correct answer along with an explanation. After constructing these QA examples, they are combined with the original data for fine-tuning.

By integrating these two strategies, we propose **R**ationale and **A**uxiliary **T**ask **E**nhanced **F**ine-**T**uning (RATE-FT) (Figure 1). RATE-FT requires the model to systematically analyze and explain its judgments and allows the model to benefit from complementary learning perspectives, reinforcing its understanding of claims through diverse yet interconnected tasks. Following the experimental setup described in Section 4, we show the comparison between RATE-FT and baseline approaches

Dataset	Method		
	Fine-Tuning	*w.o.* aux	RATE-FT
LongFact	76.1	77.5	**79.6**
Biography	78.2	79.4	**80.9**

Table 3: Results of different ablations.

Model	Method				
	$\text{Prompt}_{\text{TF}}$	$\text{Prompt}_{\text{CoT-TF}}$	Probing	Fine-Tuning	RATE-FT
Llama-3.1-70B-Instruct	73.2	76.8	79.4	80.6	**83.8**
Mistral-7B-Instruct	61.8	64.1	68.4	70.8	**73.4**
Qwen2.5-7B-Instruct	72.8	75.5	77.0	78.4	**81.1**

Table 4: Results using different models.

in Table 2, which demonstrates the superiority of RATE-FT across different datasets (see Appendix A.6 for an analysis of the effect of additional data augmentation compared to the auxiliary QA task).

5.1 Further Analysis

Ablation Study We analyze the contribution of different components of RATE-FT by investigating the variant of RATE-FT without the auxiliary task (*w.o.* aux). Table 3 presents the performance of different methods, highlighting that each component plays an important role in achieving the overall performance.

Generalization to Different Models Our experiments and analysis so far use Llama-3-8B-Instruct as the backbone model. To verify whether the performance gain of RATE-FT is consistent across different backbone models, we extend the experiments to Llama-3.1-70B-Instruct (Dubey et al., 2024), Mistral-7B-Instruct (Jiang et al., 2023), and Qwen2.5-7B-Instruct (Yang et al., 2024) on LongFact (see Appendix A.7 for details on data collection). From the results shown in Table 4, we can observe that RATE-FT consistently outperforms baseline approaches across all models, demonstrating its robustness and generalizability to diverse model architectures and scales.

In addition, we provide results of incorporating uncertainty for hallucination detection, all prompts used in our experiments, and implementation details in Appendix A.8 ~ A.12, respectively.

6 Conclusion

In this work, we systematically investigate reference-free hallucination detection in open-domain long-form generation. Our study begins with an analysis of the model's internal states, demonstrating that these states alone cannot reliably detect hallucinations. We then evaluate several existing approaches, including prompting, probing, and fine-tuning, with fine-tuning emerging as the most effective method. Building on these findings, we introduce Rationale and Auxiliary Task Enhanced Fine-Tuning (RATE-FT), a novel approach that leverages rationales and an auxiliary task to achieve significant improvements in detection performance across two datasets and various LLMs.

Limitations

One limitation of our work is its focus solely on improving the performance of the hallucination detector. A potential improvement could be to explore leveraging the detector's feedback as a reward signal to guide LLMs to generate more factual responses. Additionally, developing a more comprehensive benchmark for hallucination detection in open-domain long-form generation that covers a broader range of domains would further enhance its applicability.

References

David Paul Ausubel. 2012. *The acquisition and retention of knowledge: A cognitive view*. Springer Science & Business Media.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2024a. Complex claim verification with evidence retrieved in the wild. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3569–3587, Mexico City, Mexico. Association for Computational Linguistics.

Lida Chen, Zujie Liang, Xintao Wang, Jiaqing Liang, Yanghua Xiao, Feng Wei, Jinglei Chen, Zhenghong Hao, Bing Han, and Wei Wang. 2024b. Teaching large language models to express knowledge boundary from their own signals. *arXiv preprint arXiv:2406.10881*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9367–9385, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.

Minda Hu, Bowei He, Yufei Wang, Liangyou Li, Chen Ma, and Irwin King. 2024. Mitigating large language model hallucination with faithful finetuning. *arXiv preprint arXiv:2406.11267*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843, Singapore. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Cheongwoong Kang and Jaesik Choi. 2023. Impact of co-occurrence on factual knowledge of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7721–7735, Singapore. Association for Computational Linguistics.

Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2024. Unfamiliar finetuning examples control how language models hallucinate. *arXiv preprint arXiv:2403.05612*.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. 2024. Large language models must be taught to know what they don't know. *arXiv preprint arXiv:2406.08391*.

Sheng-Chieh Lin, Luyu Gao, Barlas Oguz, Wenhan Xiong, Jimmy Lin, Wen-tau Yih, and Xilun Chen. 2024. Flame: Factuality-aware alignment for large language models. *arXiv preprint arXiv:2405.01525*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Matéo Mahaut, Laura Aina, Paula Czarnowska, Momchil Hardalov, Thomas Müller, and Lluis Marquez. 2024. Factual confidence of LLMs: on reliability and robustness of current estimators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4554–4570, Bangkok, Thailand. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Yasumasa Onoe, Michael Zhang, Eunsol Choi, and Greg Durrett. 2022. Entity cloze by date: What LMs know about unseen entities. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 693–702, Seattle, United States. Association for Computational Linguistics.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024a. Unsupervised real-time hallucination detection based on the internal states of large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14379–14391, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024b. Unsupervised real-time hallucination detection based on the internal states of large language models. *arXiv preprint arXiv:2403.06448*.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. Minicheck: Efficient fact-checking of llms on grounding documents. *arXiv preprint arXiv:2404.10774*.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2024. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jerry Wei, Da Huang, Yifeng Lu, Denny Zhou, and Quoc V Le. 2023. Simple synthetic data reduces sycophancy in large language models. *arXiv preprint arXiv:2308.03958*.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, et al. 2024. Long-form factuality in large language models. *arXiv preprint arXiv:2403.18802*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yakir Yehuda, Itzik Malkiel, Oren Barkan, Jonathan Weill, Royi Ronen, and Noam Koenigstein. 2024. InterrogateLLM: Zero-resource hallucination detection in LLM-generated answers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9333–9347, Bangkok, Thailand. Association for Computational Linguistics.

Xiaokang Zhang, Zijun Yao, Jing Zhang, Kaifeng Yun, Jifan Yu, Juanzi Li, and Jie Tang. 2024a. Transferable and efficient non-factual content detection via probe training with offline consistency checking. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12348–12364, Bangkok, Thailand. Association for Computational Linguistics.

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng. 2024b. Self-alignment for factuality: Mitigating hallucinations in LLMs via self-evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1965, Bangkok, Thailand. Association for Computational Linguistics.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Yiran Zhao, Jinghan Zhang, I Chern, Siyang Gao, Pengfei Liu, Junxian He, et al. 2023. Felm: Benchmarking factuality evaluation of large language models. *Advances in Neural Information Processing Systems*, 36.

Yaowei Zheng, Richong Zhang, Junhao Zhang, YeYanhan YeYanhan, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

A Appendix

A.1 Benchmark Construction Details

For each prompt in the sampled subset (200 prompts), we obtain a long-form response from Llama-3-8B-Instruct with greedy decoding. Following Wei et al. (2024), we employ the model to decompose long-form responses into atomized claims and assess whether each claim is relevant to answering the corresponding prompt. For each relevant claim, we use the model to generate multi-step Google Search queries and reason about whether the search results support the claim. Claims supported by the search results are labeled as "factual", while those contradicted by the results are categorized as "hallucinated". After construction, we obtain 2394 factual claims and 223 hallucinated claims, respectively. We then randomly selected an equal number (223) of factual and hallucinated claims for experiments.

A.2 Hallucination Detection Results using Internal States

We show the hallucination detection results using different internal states in Figure 3 ~ 5

A.3 Detailed Comparison with Findings in SelfCheckGPT

(*i*) While SelfCheckGPT (Manakul et al., 2023) explores several internal states of LLMs, our work covers a broader range of variants. As illustrated in Section 3, we examine the arithmetic and geometric averages (perplexity) of all tokens, the average of tokens with the top-K lowest probability or highest entropy ($K = 1, 3, 5$), and the average of tokens with the top-$P\%$ lowest probability or highest entropy ($P = 5, 10, 15$). In contrast, SelfCheckGPT only examines the arithmetic average of all tokens and the average of tokens with the top-1 lowest probability or highest entropy.

(*ii*) Our findings differ significantly from those reported in SelfCheckGPT. While SelfCheckGPT suggests that LLM probabilities correlate well with factuality, our experiments demonstrate that neither internal state reliably, *i.e.,* better than random

guessing, predicts the correctness of a given claim. One possible explanation for this is the presence of many insignificant tokens, such as stop words, within the claim. To address this, we further investigate variants that focus only on output tokens related to entities (Appendix A.2), and the results exhibit similar patterns. Importantly, our findings are consistent with those in Kapoor et al. (2024).

A.4 Out-of-Distribution Results

We verify the effectiveness of fine-tuning in Out-of-Distribution (OOD) scenarios by training the model on LongFact and evaluating its performance on Biography. The results reported in Table 5 demonstrate that fine-tuning effectively generalizes to OOD scenarios.

A.5 More Analysis on Auxiliary Task

Comparison with F2 F2 (Hu et al., 2024) also integrates rationales and auxiliary tasks into the training process. However, its main goal is to enhance the faithfulness of model responses while we focus on improving the accuracy of hallucination detection.

Further Clarification on Motivation The underlying motivation for introducing the auxiliary question answering (QA) task into fine-tuning is that hallucination detection and mitigation are complementary and closely related tasks. This auxiliary QA task—where a question about the key information in the claim is posed, and the model is trained to provide the correct answer—helps improve the factuality of the model's responses through supervised fine-tuning. It acts as a complementary component to the primary hallucination detection task, offering the model an alternative yet closely related perspective, thereby enhancing its generalization capabilities.

A.6 Additional Data Augmentation versus Auxiliary QA Task

To isolate the effect of additional data augmentation versus the auxiliary QA task, we design two variants: (*i*) we paraphrase the original claim using GPT-4 for data augmentation and fine-tune the model on the combined data, referred to as Fine-Tuning$_{\text{para}}$, which has roughly the same amount of training data as RATE-FT; and (*ii*) we reduce the training data for RATE-FT by half (approximately the same amount as Fine-Tuning), referred to as RATE-FT$_{\text{half}}$. We conduct experiments on LongFact using Llama-3-8B-Instruct and present the results in Table 6 and 7, which demonstrate that the performance improvement primarily comes from our designed auxiliary task, rather than from additional data augmentation.

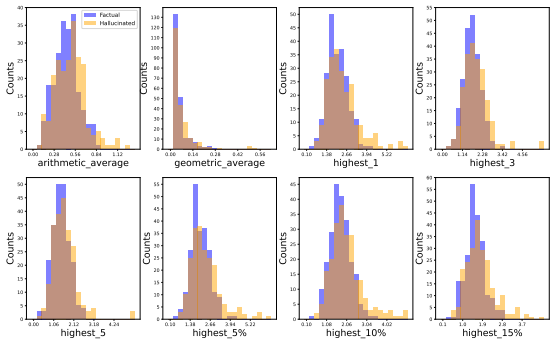


Figure 3: Hallucination detection results based on token entropy (uncertainty).

Prompt$_{\text{TF}}$	Prompt$_{\text{Prob}}$	SelfCheckGPT	Probing	Fine-Tuning
72.3	56.3	71.9	71.1	**74.7**

Table 5: Results of different methods in OOD scenarios.

A.7 Data Collection Process for Other Models

When conducting experiments using other models, we follow the exact same settings as those used for Llama-3-8B-Instruct. Specifically, for each prompt, we obtain a long-form response from the model under investigation with greedy decoding. Following Wei et al. (2024), we employ the model to decompose long-form responses into atomized claims and assess whether each claim is relevant to answering the corresponding prompt. For each relevant claim, we use the model to generate multi-step Google Search queries and reason about whether the search results support the claim. Claims supported by the search results are labeled as "factual", while those contradicted by the results are categorized as "hallucinated".

Our constructed benchmarks align well with Su et al. (2024b), as both include responses and internal states from various LLMs. The key difference is that the LLMs we investigate are all modern models (Llama-3-8B-Instruct, Llama-3.1-70B-Instruct, Mistral-7B-Instruct, and Qwen2.5-7B-Instruct), whereas the models used in Su et al. (2024b) are relatively outdated (such as LLaMA-2 and GPT-J).

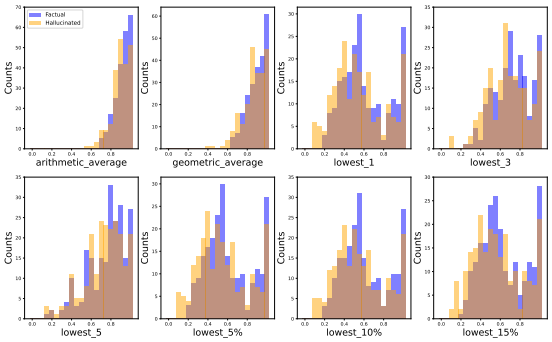


Figure 4: Hallucination detection results based on the probability of entity-related tokens.

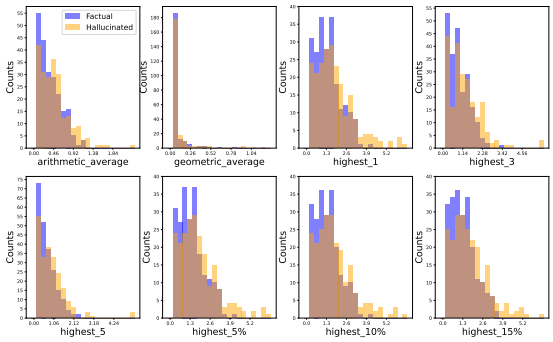


Figure 5: Hallucination detection results based on the entropy of entity-related tokens.

Fine-Tuning$_{\text{para}}$	RATE-FT
76.8	**79.6**

Table 6: Comparison between Fine-Tuning$_{\text{para}}$ and RATE-FT.

Fine-Tuning	RATE-FT$_{\text{half}}$
76.1	**78.5**

Table 7: Comparison between Fine-Tuning and RATE-FT$_{\text{half}}$.

Prompt$_{\text{CoT-TF}}$	Probing	Fine-Tuning	RATE-FT
80.4	81.1	82.4	**85.0**

Table 8: BAcc-unknown (%) of different methods on Longfact with Llama-3-8B-Instruct.

Figure 6: Prompt for extracting the original output given an atomized claim.

A.8 Incorporating Uncertainty for Hallucination Detection

To enhance hallucination detection, we propose incorporating model uncertainty into the detection process, enabling a hybrid pipeline that combines the strengths of the model and external tools. Specifically, when the model is uncertain about whether a claim is factual or hallucinated, we leverage external tools to handle ambiguous cases, improving overall performance. The process involves setting two thresholds, α_{low} and α_{high}, for classification. A claim is classified as 'factual' if $P_{\text{factual}} > \alpha_{high}$ and 'hallucinated' if $P_{\text{factual}} < \alpha_{low}$. Claims falling between these thresholds are classified as 'unknown' and delegated to external tools for further evaluation. Assuming the external tools' output is the ground truth, predictions classified as 'unknown' are treated as correct. To evaluate the hybrid pipeline, we define the BAcc-unknown metric as follows:

$$\text{BAcc-unknown} = \frac{1}{2}\Big(\frac{\#\text{ Correct Factual Predictions}}{\#\text{ Total Factual Claims}} + \frac{\#\text{ Correct Hallucinated Predictions}}{\#\text{ Total Hallucinated Claims}}\Big) \quad (1)$$

The optimal thresholds, α_{low} and α_{high}, are determined through a search on the validation set. This process ensures that BAcc on the validation set exceeds 70%, while also maximizing BAcc-unknown. The goal is to strike a balance between performance and efficiency by achieving high BAcc-unknown without generating an excessive number of 'unknown' predictions, which could substantially increase detection costs. We conduct experiments on LongFact using Llama-3-8B-Instruct and report the results in Table 8, which demonstrate that incorporating model uncertainty greatly enhances hallucination detection, as evidenced by the BAcc-unknown metric's superior performance compared to standard BAcc in resolving ambiguous cases. Moreover, RATE-FT continues to outperform all other methods with respect to the BAcc-unknown metric, highlighting its robustness and effectiveness.

A.9 Prompt for Output Extraction

After decomposition, the atomized claims may differ from the original expression in the response. To address this, we use the prompt shown in Figure 6 to retrieve the original output corresponding to a given atomized claim.

Prompt$_{TF}$

Your task is to determine the correctness of the given claim. When presented with a claim, reply with 'True' or 'False'. Make sure that your response is exactly 'True' or 'False' without any extra commentary whatsoever.

Claim: {claim}
Response:

Prompt$_{CoT-TF}$

Your task is to determine the correctness of the given claim. When presented with a claim, first explain the solution and then enclose the ultimate answer ('True' or 'False') in \boxed{}.

Claim: {claim}
Response:

Prompt$_{Prob}$

Your task is to provide the probability that the given claim is correct. When presented with a claim, reply with a number between 0.0 and 1.0. Make sure that your response is exactly a number between 0.0 and 1.0 without any extra commentary whatsoever.

Claim: The sun rises in the east and sets in the west.
Response: 1.0

Claim: Humans have four arms and three heads.
Response: 0.0

Claim: The human nose can detect over 1 trillion different scents.
Response: 0.82

Claim: The next president of South Korea will be a woman.
Response: 0.29

Claim: {claim}
Response:

SelfCheckGPT

Context: {context}

Sentence: {sentence}

Is the sentence supported by the context above? Answer Yes or No.

Answer:

Figure 7: Prompts for different prompting methods.

Prompt for 'label-rationale' Format

Your task is to determine the correctness of the given claim. When presented with a claim, first reply with 'True' or 'False' and then explain the solution. Make sure that your response starts with 'True' or 'False'.

Claim: {claim}
Response: {True/False}. {explanation}

Prompt for Question Answering

Answer the following question and provide the explanation.

Question: {question}
Answer: {answer}
Explanation: {explanation}

Prompt for Question Generation (Correct Claim)

Given a correct claim and why it is correct, first identity the key information in the claim, then transform it into a question and a correct answer (keep the answer as concise as possible) about the key information, finally give the explanation (keep it different from the given reason). Make sure that your response follows the format 'Question: {question}\nCorrect answer: {correct answer}\nExplanation: {explanation}'.

Correct claim: {correct claim}
Reason: {reason}
Response:

Prompt for Question Generation (Wrong Claim)

Given a wrong claim and why it is wrong, first identity the key information in the claim, then transform it into a question and a correct answer (keep the answer as concise as possible) about the key information, finally give the explanation (keep it different from the given reason). Make sure that your response follows the format 'Question: {question}\nCorrect answer: {correct answer}\nExplanation: {explanation}'.

Wrong claim: {wrong claim}
Reason: {reason}
Response:

Figure 8: Prompts for different components of RATE-FT.

A.10 Prompts for Baseline Approaches

Figure 7 illustrates the prompts used for different prompting methods. The prompt used for constructing training data in Probing and Fine-Tuning is the same as the prompt employed by the Prompt$_{TF}$ method.

A.11 Prompts Used in RATE-FT

Figure 8 presents all the prompts used in RATE-FT.

A.12 Implementation Details

For Prompt$_{TF}$ and Prompt$_{Prob}$, we obtain the response from the model with greedy decoding. Following Manakul et al. (2023), we set the temperature to 1.0 and generate 20 additional responses for SelfCheckGPT.

We evaluate 4 different types of contextualized embeddings for Probing: (1) the final token from the last layer (type$_1$), (2) the average of all tokens in the last layer (type$_2$), (3) the average of the final token across all layers (type$_3$), and (4) the average of type$_1$ and type$_2$ (type$_4$). The optimal embedding type, along with other hyperparameters, *e.g.*, learning rate, is selected through a search on the validation set. For Fine-Tuning and RATE-FT, we leverage the LLaMA-Factory library (Zheng et al., 2024) and perform a search on the validation set for important hyperparameters.

WiCkeD: A Simple Method to Make Multiple Choice Benchmarks More Challenging

Ahmed Elhady[1] **Eneko Agirre**[1] **Mikel Artetxe**[1,2]
[1]HiTZ Center, University of the Basque Country (UPV/EHU) [2]Reka AI
{ahmed.salemmohamed,e.agirre,mikel.artetxe}@ehu.eus

Abstract

We introduce WiCkeD, a simple method to increase the complexity of existing multiple-choice benchmarks by randomly replacing a choice with *"None of the above"*, a method often used in educational tests. We show that WiCkeD can be automatically applied to any existing benchmark, making it more challenging. We apply WiCkeD to 7 popular benchmarks and use it to evaluate 18 open-weight LLMs. The performance of the models drops 12.1 points on average with respect to the original versions of the datasets. When using chain-of-thought on 3 MMLU datasets, the performance drop for the WiCkeD variant is similar to the one observed when using the LLMs directly, showing that WiCkeD is also challenging for models with enhanced reasoning abilities. WiCkeD also uncovers that some models are more sensitive to the extra reasoning required, providing additional information with respect to the original benchmarks. We release our code and data at https://github.com/ahmedselhady/wicked-benchmarks.

1 Introduction

Multiple choice question (MCQ) benchmarks are widely used to evaluate Large Language Models (LLMs). This format consists of a question and a limited set of options, which include a correct (or best) answer and several distractors that are either incorrect or less appropriate (see Figure 1). There are various MCQ datasets that focus on different capabilities, including factual knowledge and reasoning as in MMLU (Hendrycks et al., 2021) and Arc-challenge (Clark et al., 2018), common sense as in Commonsense-QA (Talmor et al., 2019), truthfulness as in TruthfulQA (Lin et al., 2022), and domain-specific knowledge (Alonso et al., 2024; Hosseini et al., 2024). Unfortunately, most of these benchmarks got quickly saturated in the recent era dominated by LLMs, motivating harder datasets to better gauge the abilities of newer models. However, developing benchmarks is a laborious and expensive process.

Motivated by this, several recent works have explored strategies to make existing benchmarks harder, which can serve as an alternative to creating new benchmarks from scratch. For example, Gema et al. (2024) identified erroneous questions in the MMLU benchmark, and re-annotated 3k questions to be harder and more robust. Similarly, Wang et al. (2024) presented MMLU-Pro, a harder version of the MMLU benchmark that replaces noisy questions with harder ones and expands the number of distractors to include more plausible yet incorrect ones. While increasing the number of distractors reduces the probability of correct guesses by chance, creating plausible and coherent distractors is challenging and often requires manual verification (McIntosh et al., 2024).

In this work, we propose a simple yet effective method to make existing benchmarks more challenging without the need to add distractors. Namely, we present the **Wi**ld-**C**ard **D**istractor (WiCkeD) which creates a variant of any existing MCQ benchmark by keeping the question unchanged, and randomly replacing one of the choices with a wild-card distractor, *None of the above* (see Figure 1). We create WiCkeD variants of 7 popular benchmarks, and use them to evaluate 18 open-weight LLMs varying in size, model family, and training recipe. The WiCkeD datasets suffer a performance drop of 7.2-19.7 points with respect to the original datasets, depending on the model being evaluated. Using chain-of-thought does not prevent the drop (1.4-14.6), showing that WiCkeD can be used to assess reasoning capabilities. The large variance across models shows that WiCkeD is not only challenging, but it also uncovers differences in model capabilities that are not captured by the original benchmarks.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1183–1192
July 27 - August 1, 2025

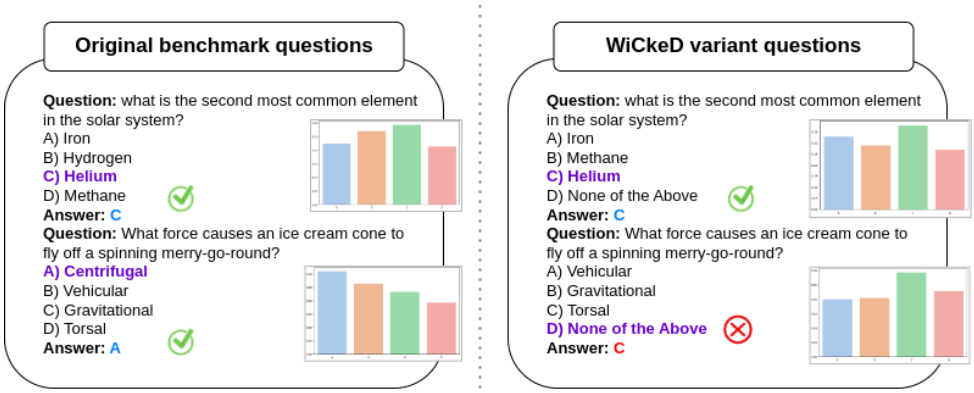


Figure 1: Two samples from MMLU-Pro (left) and its WiCkeD variant (right), where *Hydrogen* and *Centrifugal* were removed. Correct answers in bold. Llama-3.1 8B correctly answers both original questions but fails on the WiCkeD variant for the second question. The probability distribution of the model for each answer is also shown.

2 Related Work

2.1 Challenges in LLMs MCQ Benchmarks

Several works raised concerns about the effectiveness of MCQ benchmarks in LLM assessment. For example, Balepur et al. (2024) showed that some LLMs can answer MCQs using only the answer choices, without seeing the questions, and perform well-above baselines. Furthermore, more works suggested that LLMs are biased towards certain answer keys (A/B/C/D) due to unbalanced prior probabilities rather than actual knowledge (Myrzakhan et al., 2024; Clark et al., 2018).

Another line of research attributes LLMs hallucinations to being unable to identify when they lack sufficient knowledge about the subject matter (Li et al., 2024; Ji et al., 2022). Nonetheless, current evaluation benchmarks do not assess this capability effectively. A third line argues that the overestimation in performances of models is due to contamination (i.e. the model is trained on the test sets of the benchmarks), leading to wrong conclusions (Golchin and Surdeanu, 2024; Dong et al., 2024; Sainz et al., 2023). We view our work as an addition towards efficient evaluation of LLMs to avoid spurious correlations and account for knowledge and reasoning gaps.

2.2 None of the Above in Educational Tests

Multiple-choice questions (MCQs) are effective assessments when they include plausible distractors, as they encourage deeper processing to think not only about why a given choice is correct, but also why other choices are wrong and improve knowledge recall (Little et al., 2019; Little and Bjork, 2015). The use of *None of the above* as a distractor in MCQs is an area of research and debate. It can provide unique insight into the understanding of the examinees and potentially differentiate their abilities (David DiBattista and Fortuna, 2014; Dochy et al., 2001). However, *None of the above* can affect the confidence of the examinee, leading them to avoid selecting *None of the above* as the correct answer, even when it is true (Little, 2023; Odegard and Koen, 2007). Nevertheless, incorporating *None of the above* into practice tests can enhance the learning process by encouraging deeper engagement with the material (David DiBattista and Fortuna, 2014; Pezeshkpour and Hruschka, 2024; Zheng et al., 2024).

3 Methodology

We propose a method to automatically create a more challenging version of any existing MCQ benchmark without requiring any manual annotation. The difficulty of MCQ has been linked to the reasoning necessary to discriminate between competing options (McIntosh et al., 2024; Wang et al., 2024). We hypothesize that detecting the absence of the correct answer within the provided options is more challenging than selecting the correct one. To that end, we propose to add a wild-card choice *None of the above*. Note that adding *None of the above* as an additional option would not make sense, as the correct answer is always the correct option, we thus propose to replace one of the options instead.

3.1 The WiCkeD Algorithm

Given a benchmark that consists of M examples where each has N choices (one correct answer and $N-1$ distractors), we uniformly sample one option

Question: Which definition best describes media convergence?
A) The conglomeration of media outlets into large corporations.
B) The many differing views and cultures expressed in modern media.
C) Information being delivered in different formats via various digital channels.
D) merging of different forms of media into a unified digital format.

Question: Which definition best describes media convergence?
A) The conglomeration of media outlets into large corporations.
B) The many differing views and cultures expressed in modern media.
C) Information being delivered in different formats via various digital channels.
D) None of the above.

Figure 2: Applying WiCkeD on a single best answer (SBA) example (best answer D, second best answer A) would lead to an incoherent WiCkeD variant (incorrectly having *None of the above* as the gold correct answer instead of A). We thus copy SBA examples verbatim, see § 3.2 for details.

to be omitted, and append the wildcard option *None of the above*. When the correct option is replaced, the new correct option is *None of the above*. When a distractor option is replaced, the correct option continues to be correct. Figure 1 shows the result of applying WiCkeD to two examples. The goal is to produce a variant for each benchmark that contains the same number of M examples.

3.2 Coherence of WiCkeD Examples

The above algorithm does not always produce coherent examples. In some cases, there are more than one correct candidate, but only one of them is the most appropriate (see Figure 2, where D is the best answer and A is the second best answer). With the above procedure, when the replaced option is the correct one (e.g. option D in the figure), the WiCkeD variant would add *None of the above* and take this option as the correct one. However, this would be incoherent, because having removed D, A becomes the next best option. We call these examples Single Best Answer (SBA) as opposed to Single Correct Answer (SCA, where the distractors are all incorrect). As we want to keep the same number of examples we avoid adding *None of the above* to SBA examples and copy them unchanged to the WiCkeD variant of the benchmark.[1]

In order to train an example classifier to detect SBA examples, we selected four representative benchmarks (MMLU, MMLU-Pro, TruthfulQA and Commonsense-QA), sampled 4000 examples, and split them into evaluation (25%) and train (75%). We used GPT-4o-mini to automatically label the examples as SBA or SCA, and further annotated the evaluation split manually. Given the cost and slow speed of GPT-4o-mini, we used the synthetic labels to train a classifier based on BERT[2] (Devlin et al., 2019).

The recall on SBA examples for the classifier is over 98.9%, showing that we are able to detect nearly all SBA examples, and would thus have 1.1% noisy WiCkeD examples (that is, examples in the benchmark that have *None of the above* as the correct option even if a correct option exists). See Appendix A for more details about the training and evaluation procedure.

4 Experimental Setup

4.1 Benchmarks

We apply WiCkeD to six popular MCQ benchmarks that assess the knowledge, language comprehension, reasoning, and truthfulness of LLMs: MMLU, MMLU-Pro, MMLU-Redux, GPQA, CommonsenseQA, Truthful-QA, and Arc-challenge. To ensure reproducibility, we use Eval-Harness (Gao et al., 2024). Given that the selection of the option to be replaced is random, we generate five WiCkeD variants for each benchmark, and report mean and standard deviation.

Regarding the amount of SBA examples, MMLU, MMLU-Redux and MMLU-pro have the largest amount (~ 20%), with the rest of the benchmarks having less than 5% (see Appendix A). SBA examples are copied verbatim to the WiCkeD variants, but the fact that at least 80% of the examples are effectively altered makes the WiCkeD variants significantly more challenging, as we will see. Other benchmarks have less than 5% SBAs; we also leave them unchanged.

4.2 Models

We evaluate WiCkeD on 18 open-weight models covering different families and sizes. Namely, we evaluate the base and instruction-tuned models of Qwen2.5 7B, 14B and 72B (Qwen et al., 2025), Llama3.1 8B and 70B (Grattafiori et al., 2024), Gemma2 9B and 27B (Riviere et al., 2024), and Mistral-7B (Jiang et al., 2023). We also selected two DeepSeek-R1 models for their improved reasoning capabilities: distill-Lllama3.1-8B and distill-Qwen7 (DeepSeek-AI et al., 2025).

The LLM models are evaluated on the benchmarks following the standard multiple-choice prompting procedure (Robinson et al., 2023), see

[1] Excluding the SBA questions amplifies the impact of WiCkeD yet adds no other signals about performance, therefore, we leave them unchanged.

[2] https://huggingface.co/ahmedselhady/bert-base-uncased-sba-clf

Model	Size	IT	Original	WiCkeD	Δ
DS-R1-Llama	8B	-	56.6	48.6	-7.9 ±1.1%
DS-R1-Qwen	7B	-	60.8	53.4	-7.3 ±1.6%
Llama-3.1	8B	-	61.4	52.2	-9.2 ±1.7%
	8B	✓	66.0	55.0	-11.0 ±0.9%
	70B	-	76.8	67.0	-9.8 ±2.1%
	70B	✓	77.1	64.5	-12.6 ±1.3%
Mistral	7B	-	59.8	46.5	-13.2 ±1.2%
	7B	✓	59.0	47.2	-11.8 ±1.1%
Qwen-2.5	7B	-	74.7	54.9	-19.7 ±1.5%
	7B	✓	73.5	59.0	-14.5 ±1.3%
	14B	-	78.9	66.3	-12.6 ±2.1%
	14B	✓	78.9	66.6	-12.3 ±1.8%
	72B	-	84.6	72.6	-12.0 ±0.9%
	72B	✓	82.6	69.3	-13.3 ±1.0%
Gemma-2	9B	-	67.3	56.3	-10.9 ±1.2%
	9B	✓	73.3	57.6	-15.7 ±1.2%
	27B	-	68.0	54.6	-13.4 ±2.0%
	27B	✓	74.8	61.9	-12.9 ±2.3%
Average			**70.8**	**58.5**	**-12.2** ±1.5%

Table 1: Average performance on original and WiCkeD variants of the six benchmarks. IT: instruction-tuned. **Δ**: degradation from original performance

Appendix C. We set the number of few-shot examples to five, in order to ensure that in most cases there is at least one example where *None of the above* is the correct option.

In addition, we also evaluate the LLM models using zero-shot chain-of-thoughts prompting (CoT) on the three benchmarks commonly used to assess the reasoning capabilities of LLMs: MMLU, MMLU-Pro, MMLU-Redux, and GPQA. We also include 3 state-of-the-art closed-source models: OpenAI GPT-4o and GPT-4o-mini (OpenAI et al., 2024) and Gemini Flash 2.0.[3] We set the maximum generation length to 4096, unless limited by the model itself.

5 Results and Discussion

5.1 Main Results

Table 1 shows the mean accuracy of the models on the original and WiCkeD benchmarks, with a significant drop in performance. Qwen2.5-7B suffers the largest degradation (19.73%), while its DeepSeek-R1 distilled version (DeepSeek-R1-Qwen7B) suffers the least (7.35%). This suggests that models with better reasoning capabilities, like R1, are better equipped to deal with the added complexity.

Prominently, the WiCkeD variants shuffle the ranking of models. For example, the Qwen2.5-7B and Qwen2.5-7B-IT models originally performed close to the Llama-3.1-70B model. However, on the WiCkeD variants, they lag behind it by 12.1% and 8%, respectively. Similar patterns can be seen in Gemma-2-9B-IT and Gemma-2-27B-IT, which lag behind Llama-3.1-70B by 9.5% and 5.3%, respectively. Qwen2.5-72B and Llama-3.1-70B are the models that perform best in WiCkeD. There is no clear advantage from instruction-tuning, as results vary depending on the model family. Furthermore, the degradation does not decrease with model size scaling, as shown in Table 2.

Size Group	Original	WiCkeD	Δ
≤10B	63.4	51.9	-11.5
	67.9	54.7	-13.2
10-70B	73.5	60.5	-13.0
	76.8	64.2	-12.6
≥70B	80.7	69.8	-10.9
	79.8	66.9	-12.9

Table 2: Average performance per size group in original and WiCkeD variants of the six benchmarks using direct prompting. The degradation (Δ) does not decrease with model size scaling.

5.2 Chain-of-Thought Results

Table 3 shows the performance of the models[4] on the MMLU, MMLU-pro, MMLU-Redux, and GPQA WiCkeD benchmarks. The drop for these three benchmarks without CoT (direct columns in the table) is lower than the other three benchmarks, but applying CoT does not reduce the drop in WiCkeD variants, which stays above 5%. This is remarkable given that CoT is very effective at improving results on MMLU and related benchmarks. Instruction-tuned models experience significantly less degradation than their base models, especially when using CoT (see Appendix B for additional details). Notably, the DeepSpeed-R1 distilled models, Qwen7B and Llama3.1-8B, suffer around 2% each. Similarly, instruction-tuned Qwen2.5 7B and 14B suffer less than 2%. We hypothesize this is due to their enhanced reasoning capabilities.

As shown in Table 4, popular closed-source models also suffer a relatively large drop in the WiCkeD variants (3.9-8.2%). This suggests that state-of-the-art commercial models are also susceptible to the phenomenon studied in our work.

[3] deepmind.google/technologies/gemini/flash-lite/

[4] Due to compute constraints, we could not run CoT for the ~70B models

Model	Size	IT	Direct		CoT	
			WiCkeD	Δ	WiCkeD	Δ
DS-R1-Llama	8B	-	30.3	-4.1	80.1	-2.0
DS-R1-Qwen	7B	-	30.6	-4.3	74.9	-2.5
Llama-3.1	8B	-	39.7	-3.2	53.9	-5.8
	8B	✓	43.6	-2.7	57.2	-3.4
Mistral	7B	-	35.9	-3.4	36.3	-11.6
	7B	✓	33.5	-5.7	43.8	-4.9
Qwen-2.5	7B	-	45.5	-6.9	43.0	-14.6
	7B	✓	47.1	-5.3	55.4	-1.7
	14B	-	55.6	-3.6	61.5	-3.97
	14B	✓	56.7	-3.4	64.0	-1.4
Gemma-2	9B	-	36.1	-12.2	41.2	-8.9
	9B	✓	44.1	-9.3	56.3	-4.4
	27B	-	36.1	-10.8	59.2	-4.1
	27B	✓	51.3	-3.8	60.3	-3.8
Avg			**41.9**	**-5.6**	**56.3**	**-5.2**

Table 3: Performance on WiCkeD variants for MMLU, MMLU-pro, and MMLU-Redux with and without CoT. IT: instruction-tuned. **Δ**: degradation from the original benchmark.

6 Conclusion

In this paper, we introduced a simple automatic method to create more challenging variants from an existing MCQ benchmark. The large drop in the results shows that WiCkeD challenges the knowledge and reasoning of LLMs, as they need to identify the absence of the correct answer, even when using CoT. We showed that models with better reasoning capabilities suffer less in WiCkeD, such as the original Qwen7B and its distilled version of DS-R1. We see WiCkeD as an addition towards efficient evaluation of LLMs to avoid spurious correlations and challenge reasoning and knowledge gaps. A deeper look into why some models are more sensitive to WiCkeD than others can provide significant insights about uncovered limitations. We release all the code and data under open licenses.

Limitations

We manually confirmed the applicability of WiCkeD on some popular multiple-choice benchmarks whose questions can be categorized into SBAs and SCAs. However, for other benchmarks, WiCkeD might need further verification.

Acknowledgments

This work has been partially supported by the Basque Government (Research group funding IT1805-22). Ahmed Elhady holds a PhD grant supported by the Basque Government (IKER-GAITU project).

Model	WiCkeD	Δ
GPT-4o	81.4	-3.9
GPT-4o-mini	74.6	-8.2
Gemini Flash 2.0	80.3	-7.8
Avg	**78.8**	**-6.6**

Table 4: Performance of closed-source models on WiCkeD variants for MMLU, MMLU-pro, and MMLU-Redux using CoT. **Δ**: degradation from the original benchmark.

References

Iñigo Alonso, Maite Oronoz, and Rodrigo Agerri. 2024. Medexpqa: Multilingual benchmarking of large language models for medical question answering. *Artificial Intelligence in Medicine*, 155:102938.

Nishant Balepur, Abhilasha Ravichander, and Rachel Rudinger. 2024. Artifacts or abduction: How do llms answer multiple-choice questions without the question? *Preprint*, arXiv:2402.12483.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Jo-Anne Sinnige-Egger David DiBattista and Glenda Fortuna. 2014. The "none of the above" option in multiple-choice testing: An experimental study. *The Journal of Experimental Education*, 82(2):168–183.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Filip Dochy, George Moerkerke, Erik De Corte, and Mien Segers. 2001. The assessment of quantitative problem-solving skills with "none of the above"-items (nota items). *European Journal of Psychology of Education*, 16:163–177.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. 2024. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12039–12050, Bangkok, Thailand. Association for Computational Linguistics.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster,

Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. A framework for few-shot language model evaluation.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken, and Pasquale Minervini. 2024. Are we done with mmlu? *Preprint*, arXiv:2406.04127.

Shahriar Golchin and Mihai Surdeanu. 2024. Time travel in LLMs: Tracing data contamination in large language models. In *The Twelfth International Conference on Learning Representations*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Pedram Hosseini, Jessica M. Sin, Bing Ren, Bryceton G. Thomas, Elnaz Nouri, Ali Farahanchi, and Saeed Hassanpour. 2024. A benchmark for long-form medical question answering. *Preprint*, arXiv:2411.09834.

Yunjie Ji, Liangyu Chen, Chenxiao Dou, Baochang Ma, and Xiangang Li. 2022. To answer or not to answer? improving machine reading comprehension model with span-based contrastive learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1292–1300, Seattle, United States. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Moxin Li, Wenjie Wang, Fuli Feng, Fengbin Zhu, Qifan Wang, and Tat-Seng Chua. 2024. Think twice before trusting: Self-detection for large language models through comprehensive answer reflection. *Preprint*, arXiv:2403.09972.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

Jeri L Little. 2023. Does using none-of-the-above (nota) hurt students' confidence? *Journal of Intelligence*, 11(8):157.

Jeri L Little and Elizabeth Ligon Bjork. 2015. Optimizing multiple-choice tests as tools for learning. *Memory & cognition*, 43:14–26.

Jeri L Little, Elise A Frickey, and Alexandra K Fung. 2019. The role of retrieval in answering multiple-choice questions. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 45(8):1473.

Timothy R. McIntosh, Teo Susnjak, Nalin Arachchilage, Tong Liu, Paul Watters, and Malka N. Halgamuge. 2024. Inadequacies of large language model benchmarks in the era of generative artificial intelligence. *Preprint*, arXiv:2402.09880.

Aidar Myrzakhan, Sondos Mahmoud Bsharat, and Zhiqiang Shen. 2024. Open-llm-leaderboard: From multi-choice to open-style questions for llms evaluation, benchmark, and arena. *Preprint*, arXiv:2406.07545.

Timothy N Odegard and Joshua D Koen. 2007. "none of the above" as a correct and incorrect alternative on a multiple-choice test: Implications for the testing effect. *Memory*, 15(8):873–885.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017, Mexico City, Mexico. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, and 178 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2023. Leveraging large language

models for multiple choice question answering. *Preprint*, arXiv:2210.12353.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, Singapore. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Preprint*, arXiv:2406.01574.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. *Preprint*, arXiv:2309.03882.

A Detecting Single Best Answer examples

To ensure the reliability of the automatic identification of single-best-answer (SBA) questions, we uniformly sample 4K questions from the MMLU, MMLU-Pro, Commonsense-QA, and Truthful-QA benchmarks, which we divide into 1K and 3K splits. We then manually annotate the 1K samples and optimize GPT-4o-mini prompt on them for best recall. Table 6 shows the prompt template for GPT-4o-mini, which we used to annotate the 4K questions. The 3K split was then used to train our Bert-based SBA classifier on them. The classifier was trained for 2 epochs, using a learning rate of 1e-04. The model was frozen, except for the last layer and the classification head.

Table 5 shows the percentages of SBA questions on the 1K split as determined by our manual annotations, GPT-4o-mini, and the SBA classifier. The classifier is the preferred one, as it is the most conservative, that is, it detects the most SBA examples, which would be copied verbatim to the WiCkeD variant of the benchmark. The evaluation figures in the table confirm this choice, as the classifier has higher recall. The small drop in precision is harmless, as it means that we will not add *None of the above* option to those examples, and will be copied verbatim. In other words, we can estimate that WiCkeD contains 1% of incoherent examples (where there is a valid option even if *None of the above* is recorded as the correct option), and 5% of examples which do not have a *None of the above* option even if we could have added it if the classifier had 100% precision. These figures confirm the high quality of the WiCkeD variants. Table 7 shows the final SBA percentages for each benchmark as determined by the classifier.

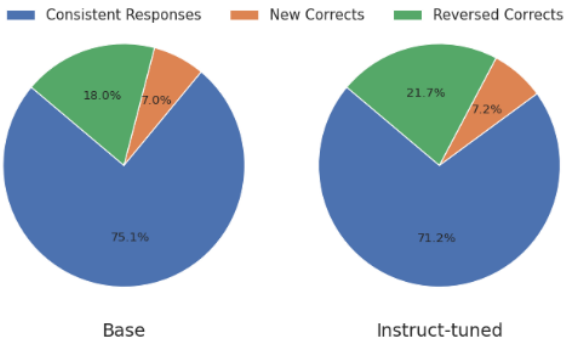


Figure 3: The changes in models' answers of the original benchmarks and the WiCkeD variant using chain-of-thoughts.

B Instruct vs Base Models on Chain-of-Thought

Results of CoT suggest the instruct models experience less degradation than their base models. To better understand why this happens, we analyze their answers. Figure 3 shows the change in answers from the original to the WiCkeD variants. Instruction-tuned models are less prone to reverse correct answers and can correct original mistakes in WiCkeD. This suggests that WiCkeD is useful for better gauging the reasoning capabilities of the models.

C Multiple Choice Prompting

In multiple choice prompting, the model is prompted with few-shot demonstrations c and a question q and the set of choices $A = \{A, B, C, D\}$. It generates a probability of the answer label $a \epsilon A$ conditioned on the prefix prompt given by:

$$\mathrm{P}(a|c,q) = \prod_{t=1}^{T} p(a_t|c, q < T) \quad (1)$$

	MMLU	MMLU-Pro	TQA	CSQA	Recall	Precision
Manual	17.3	12.3	3.3	3.8	–	–
GPT-4o-mini	18.2	13	4.2	3.9	98.5	97.4
SBA Classifier	19.6	14.2	4.5	4	98.9	95.1

Table 5: The Percentage of Single Best Answer (SBA) questions in 1K questions sampled uniformly from MMLU, MMLU-Pro, TruthfulQA (TQA), and CommonsenseQA (CSQA) as determined by our manual Annotations, GPT-4o-mini, and our trained SBA classifier. Recall and precision are computed with respect to the manual annotation.

```
"A single correct answer question is a question that can have exactly one correct answer from a given set of choices.
A single best answer question can have a most appropriate answer (for example, if this answer is omitted, another answer will be correct).
Classify the following questions into SBA and non-SBA questions. Assign a label of 1 if the question is a SBA question and a label of 0 otherwise.
Question: {question} Class:"
```

Table 6: SBA Annotation Prompt Template

MMLU	MMLU-Pro	MMLU-Redux	TruthfulQA	Commonsense QA	Arc Challenge
20.3%	16.8%	14.7%	3.2%	3.7%	5.2%

Table 7: The Percentage of Single Best Answer (SBA) questions in the benchmarks as determined by our SBA classifier. We do not apply WiCkeD to SBA questions as it can break their coherence.

The model answer is set to:

$$\underset{a \epsilon A}{\boldsymbol{argmax}}(P(a|c,q)) \quad (2)$$

```
The following are multiple-choice questions (with answers) about college computer science.

Which of the following regular expressions is equivalent to (describes the same set of strings as)
(a* + b)*(c + d)?
A. a*(c + d)+ b(c + d)
B. a*(c + d)* + b(c + d)*
C. a*(c + d)+ b*(c + d)
D. None of the above
Answer: D

A certain pipelined RISC machine has 8 general-purpose registers R0, R1, . . . , R7 and
supports the following operations.
ADD Rs1, Rs2, Rd Add Rs1 to Rs2 and put the sum in Rd
MUL Rs1, Rs2, Rd Multiply Rs1 by Rs2 and put the product in Rd
An operation normally takes one cycle; however, an operation takes two cycles if it produces a
result required by the immediately following operation in an operation sequence. Consider the
expression AB + ABC + BC, where variables A, B, C are located in registers R0, R1, R2. If the
contents of these three registers must not be modified, what is the minimum number of clock
cycles required for an operation sequence that computes the value of AB + ABC + BC?
A. 5
B. 6
C. 8
D. None of the above
Answer: B

The Singleton design pattern is used to guarantee that only a single instance of a class may be
instantiated. Which of the following is (are) true of this design pattern?
I. The Singleton class has a static factory method to provide its instance.
II. The Singleton class can be a subclass of another class.
III. The Singleton class has a private constructor.
A. I only
B. II only
C. I, II, and III
D. None of the above
Answer: C

Of the following problems concerning a given undirected graph G, which is currently known to
be solvable in polynomial time?
A. Finding the longest simple cycle in G
B. Finding the shortest cycle in G
C. Finding ALL spanning trees of G
D. None of the above
Answer:
```

Figure 4: Examples from the MMLU computer science task using WiCkeD. We show 3-shot for brevity, but 5-shot was actually used in the experiments for the main results.

Figures 4, 5, 6, and 7 show example prompts for the MMLU college computer science, Arc Challenge, Common-sense QA, and MMLU-Redux benchmarks, respectively.

```
Question: The marsh willow herb is a plant native to the northeastern United States. It grows
best in damp habitats. Which of the following environmental changes would most likely cause a
decrease in the marsh willow herb population in an area?
A. a rainstorm lasting several weeks
B. a drought lasting twelve months
C. unusually low temperatures during the month of July
D. unusually high temperatures during the month of January
Answer: B

Question: Volcanic eruptions early in Earth's history are believed to be responsible for a large
proportion of the matter now found in which Earth structure?
A. asthenosphere
B. hydrosphere
C. ozone layer
D. None of the above
Answer: D

Question: Which human activity will help decrease air pollution?
A. burning crops
B. driving a hybrid car
C. burning household garbage
D. None of the above
Answer: D

Question: Which of the following traits is most influenced by the environment?
A. Body weight
B. Eye color
C. Blood type
D. Color blindness
Answer:
```

Figure 5: Examples from the AllenAi Arc challenge using WiCkeD. We show 3-shot for brevity, but 5-shot was actually used in the experiments for the main results. The first few-shot example does not include *None of the above* option because it was classified as SBA question.

D Detailed Wicked Results

Tables 8 and 9 show the detailed performances of open-weight models using direct and CoT prompting, respectively. Since Wicked selects a random choice to be replaced every time, results may slightly vary. The reported results use a random seed of 5331. Table 10 shows the detailed performances of Closed-source models.

Model	Size	IT	MMLU		MMLU Pro		MMLU Redux		Arc Challenge		TruthfulQA		CSQA		Average	
			Org.	Wcd.	Org.	Wcd.	Org.	Wcd.	Org.	Wcd.	Org.	Wcd.	Org.	Wcd.	Org.	Wcd.
Qwen-2.5	7B	-	74.0	48.3	54.3	45.6	67.6	43.7	89.3	71.	79.3	65.8	85.1	70.3	74.9	57.5
	7B	✓	74.4	48.8	53.8	47.8	68.2	60.2	90.1	78.1	71.7	62.17	84.5	73.7	73.9	61.8
	14B	-	79.8	70.9	58.4	55.3	73.6	64.1	94.1	68.7	82.2	71.8	84.6	74.5	78.8	67.6
	14B	✓	79.9	72.7	60.2	55.6	75.2	66.7	94.4	75.1	80.1	73.1	83.4	74.2	78.9	69.6
	72B	-	86.0	78.2	67.1	62.0	81.2	64.5	95.7	74.5	88.9	73.4	89.3	77.3	84.7	71.6
	72B	✓	87.3	80.1	68.4	60.5	81.7	67.8	95.1	80.1	90.1	74.6	89.1	79.4	85.3	73.8
Gemma-2	9B	-	70.9	57.1	48.5	37.9	65.7	57.9	88.9	75.1	62.9	54.9	76.5	64.6	68.9	57.9
	9B	✓	72.2	51.4	51.2	47.1	66.1	52.8	90.1	77.1	72.3	62.4	81.9	68.4	72.3	59.9
	27B	-	70.9	57.1	48.1	35.3	67.2	52.3	89.9	69.6	78.9	59.9	81.1	70.5	72.7	57.5
	27B	✓	76.0	64.4	53.2	42.6	70.1	57.4	91.8	73.3	81.5	70.8	83.7	69.8	76.1	63.0
Llama 3.1	8B	-	64.7	56.7	42.5	39.7	58.1	51.3	79.7	58.2	58.0	49.6	73.6	64.3	62.8	53.3
	8B	✓	67.7	60.1	46.8	43.8	62.2	53.2	82.5	59.1	63.2	53.6	78.1	66.9	66.8	56.1
	70B	-	77.9	70.1	56.4	52.8	78.4	69.8	90.6	76.6	81.7	66.5	81.8	69.8	77.8	67.6
	70B	✓	82.1	68.9	58.7	44.6	79.6	67.2	91.2	79.4	80.3	69.8	82.3	69.2	79.0	66.5
Mistral	7B	-	61.5	50.4	38.1	32.7	54.1	40.8	77.7	65.8	55.8	41.7	70.7	56.4	59.6	47.9
	7B	✓	58.7	51.9	35.6	32.8	53.7	40.4	75.6	63.3	60.9	44.9	70.4	56.6	59.2	48.3
DS-R1- Distill Llama	8B	-	55.5	48.6	34.6	30.5	51.2	43.0	71.3	61.7	62.7	47.9	64.8	59.8	56.7	48.6
DS-R1- Distill Qwen	7B	-	54.1	48.4	32.2	30.9	53.8	43.2	73.8	65.7	88.4	72.5	61.9	55.7	60.7	52.7
Average			**71.9**	**60.2**	**50.5**	**44.3**	**67.1**	**55.4**	**86.8**	**70.7**	**74.4**	**61.9**	**79.0**	**67.9**	**71.6**	**60.1**

Table 8: Detailed results for benchmarks: Original (Org.) and WiCkeD (Wcd.) variants using direct prompting.

Model	Size	IT	MMLU		MMLU Pro		MMLU Redux		GPQA		Average	
			Org.	Wcd.	Org.	Wcd.	Org.	Wcd.	Org.	Wcd.	Org.	Wcd.
Qwen-2.5	7B	-	75.3	50.3	54.3	45.6	67.6	43.7	30.2	22.2	56.9	40.5
	7B	✓	76.4	53.8	56.8	47.8	68.2	60.2	32.1	23.3	58.4	46.3
	14B	-	79.8	70.9	58.4	55.3	70.6	62.1	32.8	27.8	60.4	54.0
	14B	✓	79.9	72.7	60.2	55.6	75.2	66.7	34.78	26.7	62.5	55.4
Gemma-2	9B	-	70.4	61.1	48.5	40.9	58.7	45.9	32.4	24	52.5	42.9
	9B	✓	72.2	68.4	51.2	47.1	63.1	52.8	33.8	24.9	55.1	48.3
	27B	-	74.9	70.5	48.1	42.3	67.2	61.3	31.7	25.3	55.5	49.9
	27B	✓	77.4	72.6	53.2	46.6	70.1	64.8	32.4	27.8	58.3	52.9
Mistral	7B	-	58.7	47.8	35.6	32.7	53.7	40.8	24.7	17.8	43.2	34.8
	7B	✓	61.5	51.9	38.1	32.8	54.1	42.4	28.9	20.2	45.7	36.8
Llama 3.1	8B	-	69.8	62.7	45.7	42.5	62.1	56.1	28.9	21.4	51.6	45.7
	8B	✓	71.2	66.3	47.6	44.8	64.2	60.2	31.9	24.6	53.7	48.9
DS-R1- Distill Llama	8B	-	85.9	84.6	75.1	76.1	84.5	78.5	35.8	28.6	70.3	66.9
DS-R1- Distill Qwen	7B	-	80.4	78.1	73.2	70.4	74.3	72.8	36.7	29	66.2	62.6
Average			**73.8**	**65.1**	**53.3**	**48.6**	**66.7**	**57.7**	**31.9**	**24.5**	**56.4**	**49.0**

Table 9: Detailed results for the MMLU, MMLU-pro, MMLU-redux, and GPQA benchmarks' Original (Org.) and WiCkeD (Wcd.) variants using chain-of-thought prompting.

Model	MMLU		MMLU Pro		MMLU Redux		Average	
	Org.	Wcd.	Org.	Wcd.	Org.	Wcd.	Org.	Wcd.
GPT-4o	81.3	75.9	63.7	55.6	78.9	67.9	74.6	66.5
GPT-4o-mini	89.6	84.0	70.5	67.3	84.3	81.3	81.5	77.5
Gemini Flash 2.0	82.8	77.4	75.9	64.2	82.2	75.9	80.3	72.5
Average	**84.6**	**79.1**	**70.0**	**62.4**	**81.8**	**75.0**	**78.8**	**72.2**

Table 10: Detailed results of closed-source models in the MMLU, MMLU-pro, and MMLU-redux benchmarks' Original (Org.) and WiCkeD (Wcd.) variants using chain-of-thought prompting.

Question: An owl hunts mice that live in a farmer's field. After the farmer gathers the crops, the mice have fewer places to hide. Which is most likely to happen after the crops are gathered?
A. The owl will catch more mice.
B. The owl will hunt in a different field.
C. The owl will have new material to build its nest.
D. The owl will have a hard time feeding its young.
Answer: A

Question: What can a flower become?
A. a fruit
B. a leaf
C. a branch
D. None of the above
Answer: A

Question: Which statement accurately describes the neutrons in any element?
A. The number of neutrons equals the number of electrons.
B. The charge of a neutron is always negative.
C. Neutrons are more massive than electrons.
D. None of the above
Answer: D

Question: Cells take in food for energy. The part of the cell that aids in digestion of the food is the lysosome. What is the main role of lysosomes in the process of food digestion?
A. building proteins
B. breaking down wastes
C. converting energy from one form into another
D. None of the above
Answer:

Figure 6: Examples from the Common-sense QA using WiCkeD. We show 3-shot for brevity, but 5-shot was actually used in the experiments for the main results. The first few-shot example does not include *None of the above* option because it was classified as SBA question.

Protons used in cancer therapy are typically accelerated to about 0.6c. How much work must be done on a particle of mass m in order for it to reach this speed, assuming it starts at rest?
A. 0.25mc^2
B. 0.60mc^2
C. 1.25mc^2
D. None of the above
Answer: A

A resistor in a circuit dissipates energy at a rate of 1 W. If the voltage across the resistor is doubled, what will be the new rate of energy dissipation?
A. 0.25 W
B. 0.5 W
C. 4 W
D. None of the above
Answer: C

A rod measures 1.00 m in its rest system. How fast must an observer move parallel to the rod to measure its length to be 0.80 m?
A. 0.60c
B. 0.70c
C. 0.80c
D. None of the above
Answer: A

A photon strikes an electron of mass m that is initially at rest, creating an electron-positron pair. The photon is destroyed and the positron and two electrons move off at equal speeds along the initial direction of the photon. The energy of the photon was
A. mc^2
B. 2mc^2
C. 3mc^2
D. None of the above
Answer:

Figure 7: Examples from the MMLU-Redux using WiCkeD. We show 3-shot for brevity, but 5-shot was actually used in the experiments for the main results.

Cross-Lingual Representation Alignment Through Contrastive Image-Caption Tuning

Nathaniel Krasner, Nicholas Lanuzo, Antonios Anastasopoulos
Department of Computer Science, George Mason University; Fairfax, VA.
Correspondence: nkrasner@gmu.edu

Abstract

Multilingual alignment of sentence representations has mostly required bitexts to bridge the gap between languages. We investigate whether visual information can bridge this gap instead. Image caption datasets are very easy to create without requiring multilingual expertise, so this offers a more efficient alternative for low-resource languages. We find that multilingual image-caption alignment can implicitly align the text representations between languages, languages unseen by the encoder in pretraining can be incorporated into this alignment post-hoc, and these aligned representations are usable for cross-lingual Natural Language Understanding (NLU) and bitext retrieval.[1]

1 Introduction

Encoder language models are very popular and widely used for extracting semantic information from text to be used downstream for natural language understanding (NLU) tasks. In general, an encoder language model (LM) is pretrained on a large corpus using self-supervision and then a smaller component is fine-tuned on annotated data using the representations produced by the pretrained LM. For widely spoken data-rich languages, this is no problem and the existence of task-specific, annotated data is a given (Joshi et al., 2020; Blasi et al., 2022). For low-resource languages this is rarely the case, and collecting data for each task in such languages is expensive and time consuming. Thus, cross-lingual knowledge transfer is a more practical direction for low-resource languages than further data collection.

The internal representations of encoder models trained on multilingual data tend to be disjoint, so the representation of a sentence in language A may not be similar to the representation of its translation in language B. Most likely, this is the result of pretraining data imbalance and domain mismatch across the languages included in their pretraining. If these internal representations were aligned such that representations of translations *were* similar, cross-lingual transfer for NLU tasks should be much easier to achieve, as Hu et al. (2021) showed. This cross-lingual transfer of task knowledge can greatly benefit speakers of low-resource languages by giving them access to NLP tools without the difficulty of annotating task-specific data in their language. As an additional benefit, these aligned representations can be used to mine bitexts from large scraped corpora to build parallel translation datasets (Team et al., 2022).

In this work, we explore whether one could encourage multilingual representation alignment *without any parallel data*, by relying instead on images as shared modality across languages. This is a worthwhile direction to pursue for two reasons. First, parallel text curation through expert translation is time-consuming, expensive, and requires bilingual annotators. In contrast, it is easy for an annotator to describe an image to produce a caption regardless of which language(s) they speak (Madaan et al., 2020). Second, language documentation efforts often produce media accompanied with monolingual audio or text in the language of interest. Developing techniques which leverage such materials could enable the creation of technologies for these otherwise under-served languages.

To summarize, we (1) show that a multilingual text-image contrastive learning setup can produce multilingually aligned text representations; (2) focus specifically on Quechua, as an example of a language unseen during pretraining that may benefit from such approaches; and (3) show that the addition of an unseen language does not degrade representation quality in other languages.

[1]Data and code will be publicly released at https://github.com/nkrasner/cl-clip-align.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1193–1199
July 27 - August 1, 2025

2 Related Work

Previous endeavors in multilingual alignment in the absence of parallel-text supervision have predominantly concentrated on the alignment of static word-embeddings through adversarial techniques (Zhang et al., 2017; Chen and Cardie, 2018). Approaches that extend multilingual alignment to sentence-level representations have generally necessitated a bitext signal (Feng et al., 2022; Escolano et al., 2021; Artetxe and Schwenk, 2019), with limited exceptions employing adversarial methodologies (Aghajanyan et al., 2019; Tien and Steinert-Threlkeld, 2022). Even though multilingual alignment may extend to languages not encountered during fine-tuning (Tien and Steinert-Threlkeld, 2022), we hypothesize that a more direct fine-tuning strategy using some pivot (even if not textual) could potentially produce superior alignment for languages with limited bitext resources.

Contrastive methods have been used for text-text (Feng et al., 2022) encoder alignment as well as text-image encoder alignment in both monolingual (Radford et al., 2021) and multilingual (Muraoka et al., 2023; Bianchi et al., 2023) settings. One such text-image alignment work introduces an image representation into the input sequence of NLU tasks leading to improved cross-lingual transfer (Muraoka et al., 2023). This offers additional support to our hypothesis that visual information can act as a semantic bridge between languages.

3 Method, Experiments, and Results

Our approach strings together a text encoder with a vision encoder. These two produce representations for each modality input, which are then used in a contrastive learning setup. In particular, given pairs of image representations E_i and caption representations E_c we use the following, simple contrastive loss function:

$$S = E_c \cdot E_i^\top * t$$
$$L(E_i, E_c) = \text{CrossEntropy}(S, I),$$

where I is the identity matrix and t is a learned temperature parameter.

This is similar to what CLIP (Radford et al., 2021) used for text-image alignment and LaBSE (Feng et al., 2022) for text-text alignment.

3.1 Experimental Setup

Datasets We work with the MS-COCO dataset (Lin et al., 2014), which provides 118k English Image-Caption pairs. Using Google Translate, we translate the English captions into Spanish, Japanese, Hindi, and Quechua. From this 5-way parallel image caption dataset, we derive 4 datasets for various experiments:

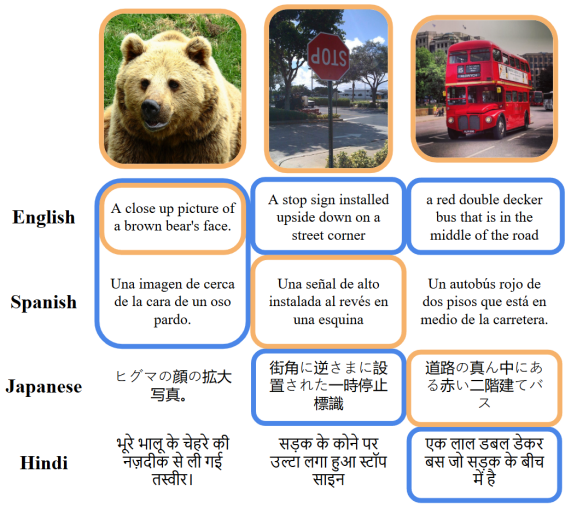


Figure 1: A demonstration of the data sampling methods. Orange boxes highlight how our multi-modal approaches sample data. Blue boxes highlight how the `Eng-Pivot` approach samples data.

1. `Eng-Pivot`: The English captions from MS-COCO paired with one translation each from a rotation of Spanish, Japanese, and Hindi.
2. `Eng-only`: The English MS-COCO dataset without translations to other languages.
3. `Multilingual`: The MS-COCO dataset but each caption is from a rotation of English, Spanish, Japanese, and Hindi with only one language paired with each image.
4. `Multilingual+Quechua`: The same as the `Multilingual` dataset but with Quechua added into the rotation of languages.

While most of these datasets are designed for use with text-image alignment, the `Eng-Pivot` dataset is used for text-text alignment to create a model similar to LaBSE (Feng et al., 2022) with a comparable data size to our other models. This is the only dataset which contains parallel text data. Figure 1 gives a visual representation of this distinction.

Training We fine-tune an XLM-Roberta-Large (XLM-R) (Conneau et al., 2020) text encoder and a VIT-Base-patch16-224-in21k (Dosovitskiy et al., 2021) image encoder for 10 epochs with early stopping.

The token-level representations are mean pooled to create a sentence-level representation. Since the

hidden dimensions of these encoders do not match, we add a linear layer to their outputs to adapt them to a matching dimensionality of 512. Following existing approaches to text-image alignment under these circumstances (Bianchi et al., 2023), we allow these linear layers to warm up for a certain number of steps before fine-tuning the encoders themselves. In our case, we chose to begin training the encoders halfway through the first epoch since the learning curves had flattened out by that point.

3.2 Experiment 1: Does multilingual text-image alignment lead to text-text alignment?

We hypothesize that text-image alignment involving multiple languages will implicitly align text representations between languages.

With the exception of the `Eng-Pivot` encoder (which is trained on bitext alignment), our encoders are only fine-tuned to align the text representations to the image representations, but we evaluate them on their alignment between text representations. Specifically, we use the FLoRes-200 dataset (Team et al., 2022), which contains 1012 204-way parallel sentences including all of our test languages. We perform a formal analysis using the task of bitext retrieval (Heffernan et al., 2022; Duquenne et al., 2023) as well as a visual analysis via t-SNE.

We compare against a baseline of the off-the-shelf XLM-R encoder, as well as one fine-tuned on text-image alignment using the English only (`Eng-Only`) dataset, and another trained directly on contrastive text-text alignment with an English pivot similarly to LaBSE (Feng et al., 2022).

For each sentence in each language, we search the English sentences in FLoRes-200 for the minimum cosine distance to find the corresponding English translation. If the true translation is selected, we count that sentence as correct. We calculate the retrieval accuracy over each language and then aggregate using the mean over all languages to produce a final score. Since XLM-R has not seen all of these languages in pretraining, we report the retrieval accuracy over the disjoint subsets of languages on which it was pretrained (or not). Table 1 contains the results.

While not quite matching the `Eng-Pivot` text-text aligned encoder, the `Multilingual` text-image aligned encoder is still very capable in the bi-text retrieval task. The `Eng-Only` text-image alignment improves on the abysmal results of the plain XLM-R model, but does not compare with the `Multilingual` alignment. This is likely because the pretraining of XLM-R does not scale to sentence level tasks well (Reimers and Gurevych, 2019). The text-image alignment, on its own, may expand the existing knowledge of XLM-R to the sentence level.

Encoder	All (203 langs)	in XLM-R (92 langs)	not in XLM-R (111 langs)	Quechua
XLM-R	0.5	0.6	0.4	0.5
Eng-Pivot	**62.2**	**92.6**	**37.1**	13.1
Eng-Only	18.3	27.5	10.7	7.2
Multilingual	55.7	82.2	33.7	18.0
+ Quechua	50.4	76.6	28.6	**29.2**

Table 1: Bitext retrieval accuracy on All of FLoRes-200, on the subset of languages in/not in XLM-R's pretraining, and just on Quechua.

To further visualize the multilingual alignment of our encoders, we generate sentence-level representations for all sentences in the FLoRes-200 dataset and use t-SNE to project them down to 2 dimensions while preserving relative distances. We plot these embeddings in Figure 2 for the 4 fine-tuning languages with lines connecting parallel cliques of translated sentences. This way we can visualize whether an encoder produces language-specific clusters or whether certain sentences are encoded far from their translations.

Figure 2 shows that the original XLM-R representations are not aligned at all. Tuning only on the English image-caption data leads to better alignment than the untuned model, but the languages still form distinct clusters. Our `Multilingual` approach falls just short of the text-text aligned model in terms of the number of misaligned translations and adding Quechua into the mix does not make it that much worse. Interestingly, the text-image aligned models have tighter inter-sentence clusters indicating that the image alignment may have drawn connections between sentences that are not captured by a text-only semantic space.

Real versus Synthetic Captions To control for external factors, our experiments rely on synthetic captions generated by translating the MS-COCO English captions. To measure the effects of this synthetic data, we trained two additional models using the English and German captions from the Multi30k dataset (Elliott et al., 2016). For the first, we alternate between these real English and German captions in the training data. We will refer to this approach as `Multi30k`. For the second, we replace the german captions from `Multi30k` with the translations of their English counterparts. We

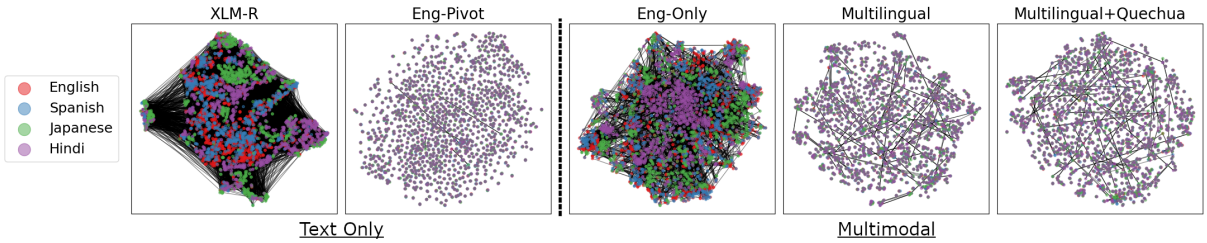


Figure 2: t-SNE embeddings for the outputs of each encoder over FLoRes-200 sentences. Translations are shown as cliques with lines connecting them. Visible lines, such as those in the XLM-R and `Eng-Only` panels, indicate that representations of translated sentences are far from each other, ie., *poor* alignment. While not as clear as parallel text alignment (`Eng-Pivot`), multilingual image-text alignment (two rightmost panels) shows promising results.

will refer to this as `Translated-Multi30k`.

Using the same bitext retrieval procedure from before, we find that the retrieval accuracy for `Multi30k` across all languages in FLoRes-200 is 40.4% and in German the retrieval accuracy is 94.6%. Similarly, `Translated-Multi30k` scores a retrieval accuracy of 38.7% across all of FLoRes-200 and 94.1% in German. Using translated captions does not significantly change the quality of the alignment.

3.3 Experiment 2: Can a language unseen in the encoder's pretraining be added using only image caption tuning?

Here, we turn to investigating the possibility of using only image-caption data to obtain good representations for a language *unseen* during pretraining, without any parallel text data. This approximates a real setting where we could ask an annotator to write image captions in a low-resource language which we want to add to our aligned language encoder for use in downstream tasks in a zero-shot cross-lingual transfer setting (Madaan et al., 2020).

We find that languages not included in the pretraining or fine-tuning still benefit from some alignment. But as one would expect, not to the same degree as those which have been already included in the model's training data.

We retrained the encoder from Experiment 1, but now with a dataset that also mixes in Quechua captions. Indigenous Latin American languages, including Quechua, are not included in the pretraining data of XLM-R. Quechua is also typologically distinct from all other pretraining languages.

We calculate the retrieval accuracy on FLoRes-200 from Quechua to English as well as the overall X→English accuracy to determine how well Quechua has been integrated into the encoder and aligned with other languages.

When Quechua is added to the image-caption dataset, the overall performance goes down slightly, but the performance on Quechua is greatly improved (cf last two rows of Table 1) from 18% to 29.2%. Importantly, the average accuracy for all other languages remains largely unaffected – we attribute the small drop in performance to the fact that we reduced the data in the other four languages to ensure experimental data-size comparability; in practice, this is not a requirement in the real world.

3.4 Experiment 3: Are the downstream qualities of the representations preserved and is cross-lingual transfer possible?

Here, we go beyond intrinsic evaluation to test our embeddings for a downstream task: natural language inference (NLI). Since images and text contain different types of semantic information, we want to ensure that aligning a text encoder to an image encoder does not overwrite the features which are useful for downstream NLU tasks.

We train simple feed-forward NLI models on frozen representations from each of the models in the previous experiments using the combined MultiNLI (Williams et al., 2018) training and dev sets.

We train using the MultiNLI train and dev datasets which only contain English samples. Any samples marked by the authors as lacking agreement were discarded. For evaluation of downstream NLI quality, we use the XNLI (Conneau et al., 2018) and AmericasNLI (Ebrahimi et al., 2022) test sets to measure both English NLI and cross-lingual transfer performance.

For each encoder, we train identical NLI models with input features ($\oplus$ stands for concatenation):

$$x_i = e(p_i) \oplus e(h_i) \oplus |e(p_i) - e(h_i)| \oplus e(p_i) * e(h_i)$$

where e is the encoder and p_i and h_i are a premise and hypothesis respectively (Conneau et al., 2017).

Encoder	XNLI															AmericasNLI											Avg
	en	es	hi	ar	bg	de	el	fr	ru	sw	th	tr	ur	vi	zh	quy	aym	bzd	cni	gn	hch	nah	oto	shp	tar		
XLM-R	50	44	44	45	43	43	43	47	44	37	42	43	42	46	44	34	33	33	35	35	34	35	32	34	33		40
Eng-Only	53	50	46	47	49	49	48	51	50	42	47	47	45	48	48	40	38	35	35	38	39	36	35	37	36		44
Eng-Pivot	**67**	**65**	**60**	**61**	**63**	**64**	**63**	**65**	**62**	**52**	**61**	**61**	**58**	**63**	**62**	39	**40**	37	**41**	40	**40**	**42**	**39**	**44**	39		**53**
Multilingual	55	52	51	51	53	52	52	53	52	45	51	51	48	52	51	37	35	36	37	37	37	37	37	39	**40**		46
+ Quechua	56	53	51	51	53	53	53	53	53	45	50	51	49	52	51	**41**	36	**39**	**41**	**41**	36	40	**39**	41	**40**		47

Table 2: Rounded XNLI and AmericasNLI accuracy. Languages seen for alignment fine-tuning are underlined. NLI models are only trained on English data with frozen encoders; results in other languages require cross-lingual transfer.

The NLI models are a simple feed-forward architecture with 2 hidden layers and a hidden size of 2048. They are trained using the Adam optimizer and a learning rate of $2 * 10^{-5}$ for 100 epochs with early stopping.

The results in Table 2 show that the alignment of the text encoder with the space of the image encoder does not damage the quality of the text representations for downstream use, but actually improves them. Comparing the `Multilingual` image aligned model before and after adding Quechua, downstream performance is somewhat uncoupled from bitext retrieval performance. The addition of Quechua matched or exceeded the performance without it across nearly all languages, suggesting that NLI performance benefits from increased language coverage regardless of individual language data size. English represents $\frac{1}{4}$ of the `Multilingual` dataset and $\frac{1}{5}$ after adding Quechua, but the addition of Quechua increased the NLI score on English! Additionally, fine-tuning the encoder on the `Eng-Only` dataset only made a minimal improvement to the XLM-R performance even though it saw the largest portion of English data.

Additionally, the Quechua captions lead to improved results across the AmericasNLI languages even matching or outperforming the `Eng-Pivot` results in many of those languages. Adding a language from an unseen family not only improves representation quality for that language, but also improves cross-lingual transfer to unseen languages in that family.

4 Conclusion

The task of multilingual text-image contrastive alignment implicitly aligns text from multiple languages into the same space. This alignment carries over into unseen languages, and performance on a particular unseen language can be improved by collecting image-caption pairs in that language.

While this technique does not outperform SOTA methods, it performs remarkably well considering the non-reliance on parallel corpora. For low resource languages, this method could act as a bootstrapping step to scrape higher quality bitexts for use in further alignment.

Limitations

With the addition of Quechua to the training set, the drop in overall bitext retrieval performance could be due to the decrease in data for the other languages to accommodate the Quechua data. Whether this is the case is not captured by our experiment, but can be taken into account in a follow-up work.

Acknowledgments

This work was partially supported by resources provided by the Office of Research Computing at George Mason University (URL: https://orc.gmu.edu) and funded in part by grants from the National Science Foundation (Award Number IIS-2327143).

References

Armen Aghajanyan, Xia Song, and Saurabh Tiwary. 2019. Towards language agnostic universal representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4033–4041, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Federico Bianchi, Giuseppe Attanasio, Raphael Pisoni, Silvia Terragni, Gabriele Sarti, and Dario Balestri. 2023. Contrastive language–image pre-training for the Italian language. In *Proceedings of the 9th Italian*

Conference on Computational Linguistics (CLiC-it 2023), pages 78–85, Venice, Italy. CEUR Workshop Proceedings.

Damian Blasi, Antonios Anastasopoulos, and Graham Neubig. 2022. Systematic inequalities in language technology performance across the world's languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5505, Dublin, Ireland. Association for Computational Linguistics.

Xilun Chen and Claire Cardie. 2018. Unsupervised multilingual word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 261–270, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of International Conference on Learning Representations*.

Paul-Ambroise Duquenne, Holger Schwenk, and Benoît Sagot. 2023. Sonar: Sentence-level multimodal and language-agnostic representations. *Preprint*, arXiv:2308.11466.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios, Ivan Vladimir Meza Ruiz, Gustavo Giménez-Lugo, Elisabeth Mager, Graham Neubig, Alexis Palmer, Rolando Coto-Solano, Thang Vu, and Katharina Kann. 2022. AmericasNLI: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6279–6299, Dublin, Ireland. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74. Association for Computational Linguistics.

Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Mikel Artetxe. 2021. Multilingual machine translation: Closing the gap between shared and language-specific encoder-decoders. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 944–948, Online. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2101–2112, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Junjie Hu, Melvin Johnson, Orhan Firat, Aditya Siddhant, and Graham Neubig. 2021. Explicit alignment objectives for multilingual bidirectional encoders. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3633–3643, Online. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

Aman Madaan, Shruti Rijhwani, Antonios Anastasopoulos, Yiming Yang, and Graham Neubig. 2020. Practical comparable data collection for low-resource languages via images. In *Practical ML for Developing Countries Workshop, ICLR*.

Masayasu Muraoka, Bishwaranjan Bhattacharjee, Michele Merler, Graeme Blackwood, Yulong Li, and Yang Zhao. 2023. Cross-lingual transfer of large language model by visually-derived supervision toward low-resource languages. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 3637–3646, New York, NY, USA. Association for Computing Machinery.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Chih-chan Tien and Shane Steinert-Threlkeld. 2022. Bilingual alignment transfers to multilingual alignment for unsupervised parallel text mining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8696–8706, Dublin, Ireland. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1959–1970, Vancouver, Canada. Association for Computational Linguistics.

LAMB: A Training-Free Method to Enhance the Long-Context Understanding of SSMs via Attention-Guided Token Filtering

Zhifan Ye[1*], **Zheng Wang**[1], **Kejing Xia**[1], **Jihoon Hong**[1], **Leshu Li**[1], **Lexington Whalen**[1], **Cheng Wan**[1], **Yonggan Fu**[1], **Yingyan (Celine) Lin**[1], **Souvik Kundu**[2]
[1]Georgia Institute of Technology, USA [2]Intel Labs, USA
{zye327,celine.lin}@gatech.edu, souvikk.kundu@intel.com

Abstract

State space models (SSMs) achieve efficient sub-quadratic compute complexity but often exhibit significant performance drops as context length increases. Recent work attributes this deterioration to an exponential decay in hidden state memory. While token filtering has emerged as a promising remedy, its underlying rationale and limitations remain largely non-understood. In this paper, we first investigate the attention patterns of Mamba to shed light on why token filtering alleviates long-context degradation. Motivated by these findings, we propose LAMB, a training-free, attention-guided token filtering strategy designed to preserve critical tokens during inference. LAMB can boost long-context performance for both pure SSMs and hybrid models, achieving up to an average improvement of 30.35% over state-of-the-art techniques on standard long-context understanding benchmarks. Our analysis and experiments reveal new insights into the interplay between attention, token selection, and memory retention, and are thus expected to inspire broader applications of token filtering in long-sequence modeling. Our code is available at https://github.com/GATECH-EIC/LAMB.

1 Introduction

State-space models (SSMs), including Mamba variants (Gu and Dao, 2024; Dao and Gu, 2024), have emerged as a sub-quadratic alternative to traditional transformers, enabling large language models (LLMs) to process long contexts more efficiently (Bai et al., 2024). However, prior studies (Waleffe et al., 2024; Azizi et al., 2025) have shown that vanilla Mamba models struggle with contexts exceeding their training length, primarily due to the exponential decay of their hidden states. This shortcoming prevents SSMs from fully realizing their sub-quadratic efficiency benefits.

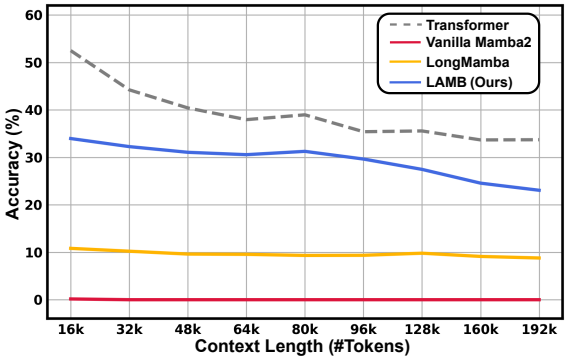


Figure 1: Comparison of average performance on RULER across various context lengths. We evaluate a Transformer baseline (Llama-3.2-1B (Llama Team, 2024)), the vanilla Mamba2-1.3B (Dao and Gu, 2024), the LongMamba-enhanced Mamba2-1.3B model (state-of-the-art baseline for training-free long-context enhancement), and the Mamba2-1.3B model enhanced with the proposed LAMB framework.

Recent works (Ben-Kish et al., 2025; Ye et al., 2025) have mitigated this issue by introducing token filtering techniques to extend Mamba models' effective context length. For instance, LongMamba (Ye et al., 2025) categorizes the hidden state channels of SSMs into global and local channels, capturing global and local information, respectively. It then filters out less important tokens from the global channels to mitigate information decay. While these methods have demonstrated effectiveness across various applications, a comprehensive understanding of their underlying mechanisms, strengths, and limitations remains lacking. In this paper, we make the following contributions:

- We present a new study of token filtering methods by analyzing the attention [1] patterns of Mamba. Specifically, we observe that a small subset of tokens dominates the attention map, indicating that preserving these key tokens

[*]Work done as a part of his internship at Intel.

[1]Note that the "attention" refers to attention-equivalent metrics derived from the original Mamba mechanism, as formulated by Ali et al. and detailed in Sec. 2.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1200–1209
July 27 - August 1, 2025

can be sufficient for retaining the major attention pattern. Furthermore, we find that an attention-guided approach can outperform previous approaches when it comes to identifying these key tokens.

- Building on these insights, we introduce LAMB, a ***training-free*** method for **L**ong-context extension driven by **A**ttention-guided token filtering in **M**am**B**a. LAMB features a custom attention metric that could identify important tokens from lengthy input sequences. Extensive experiments demonstrate that LAMB surpasses the state-of-the-art (SOTA) method on both Mamba models (as shown in Fig. 1) and hybrid architectures.

2 Preliminary and Related Works

Mamba SSM. The Mamba model (Gu and Dao, 2024; Dao and Gu, 2024) is built by stacking multiple layers of Mamba blocks, each of which processes an input sequence of L tokens while maintaining its own hidden state h. The hidden state is updated recurrently within each block as the tokens are processed sequentially. For a given channel c and token index t, the update is defined by:

$$h_{t;c} = \bar{A}_{t;c}\, h_{t-1;c} + \bar{B}_{t;c}\, x_{t;c}, \quad (1)$$

$$\bar{A}_{t;c} = \exp\big(\Delta_{t;c}\, A_c\big), \quad \bar{B}_{t;c} = \Delta_{t;c}\, B_{t;c}. \quad (2)$$

where $h_{t;c} \in \mathbb{R}^N$ denotes the hidden state vector after processing token t, $A_c \in \mathbb{R}^{N\times N}$ is a constant negative matrix and $B_{t;c} \in \mathbb{R}^{N\times 1}$ is a token-dependent matrix. $\Delta_{t;c} \in (0,1)$ is a scalar that modulates the update magnitude: When $\Delta_{t;c} \approx 0$, we have $\bar{A}_{t;c} \approx 1$ and $\bar{B}_{t;c} \approx 0$, resulting in little to no change in the hidden state, whereas larger values of $\Delta_{t;c}$ produce more significant updates. The output at token t is given by

$$y_{t;c} = C_{t;c}^{\top} h_{t;c}, \quad (3)$$

where $C_{t;c} \in \mathbb{R}^{N\times 1}$. For brevity, we will omit the subscript c in the remainder of this work unless channel-specific behavior is being emphasized.

Attention in Mamba Models. Previous work (Ali et al., 2024) introduced an attention score to quantify the contribution of jth (x_j) to the ith (y_i) token in Mamba. Specifically, unrolling Eq. 3 yields

$$y_i = \sum_{j=1}^{i} C_i^{\top}\Big(\prod_{k=j+1}^{i} \bar{A}_k\Big)\bar{B}_j x_j = \sum_{j=1}^{i} \alpha_{i,j} x_j, \quad (4)$$

where $\alpha_{i,j} = C_i^{\top}\Big(\prod_{k=j+1}^{i} \bar{A}_k\Big)\bar{B}_j$ represents the contribution of x_j to y_i, serving as an analogue to the attention weights in Transformer-based models.

Long-Context Mamba. Prior studies (Ben-Kish et al., 2025; Azizi et al., 2025) have shown that Mamba models struggle to generalize beyond the sequence lengths encountered during training. This limitation stems from the hidden state update in Eq. 1, where each new token multiplies the previous hidden state h_{t-1} by a decay factor $\bar{A}_t < 1$. As the sequence length increases, this repeated decay progressively attenuates earlier information.

The previous method, LongMamba (Ye et al., 2025), mitigates this effect by first identifying a subset of hidden state channels C_g, termed "global channels", that retain long-range information. A channel is classified as global if the cumulative decay over a sampled sequence of length S, $\Pi_{t=1}^{S}\bar{A}_t$, falls below a predefined threshold θ: $\prod_{t=1}^{S}\bar{A}_t < \theta$; such channels decay slowly and thus preserve global context. LongMamba then discards tokens whose Δ_t is small and would make only negligible updates in these channels. Concretely, it sets $\Delta_t = 0$ for every discarded token, which forces $\bar{A}_t = 1$ and $\bar{B}_t = 0$, so the update rule in Eq. 1 simplifies to $h_t = h_{t-1}$. Hence, the hidden state remains unchanged for discarded tokens, while retained tokens update it normally.

LAMB employs a similar framework: it identifies the same set of global channels and sets Δ_t to zero for discarded tokens. However, LAMB selects the tokens to discard using an attention-guided importance score rather than LongMamba's heuristic based on the magnitude of Δ_t. This principled selection yields stronger long-context performance, as demonstrated in Sec. 5.

3 Analysis of Token Filtering

In this section, we systematically investigate the role of token filtering in SSM models by closely analyzing their attention patterns. Our goal is to understand why certain tokens are more critical than others for maintaining performance on long-context tasks and how to identify these critical tokens.

3.1 Attention Patterns in SSMs

Fig. 2 (a) shows the attention map of a randomly selected global channel in the Mamba2-1.3B model with 512 input tokens. A pronounced column-wise pattern emerges, indicating that only a small subset of tokens (columns) dominates the attention distribution. To quantify this effect, we compute each token's cumulative contribution to future generated

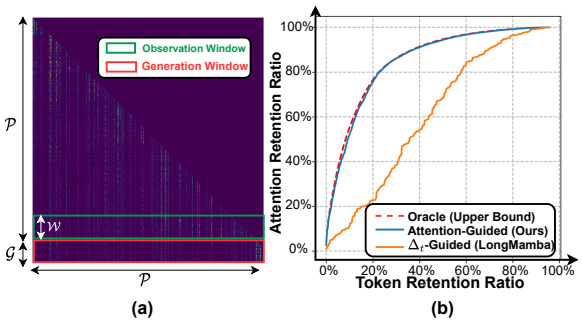


Figure 2: (a) Attention map of a randomly selected global channel in Mamba2-1.3B for an input sequence $\mathcal{P}$ of 512 tokens followed by 32 generated tokens $\mathcal{G}$. (b) Attention retention ratio within the generation window (red box in (a)) as a function of the token retention ratio under three token filtering strategies: *Oracle* (upper bound, requires oracle access to the future attention map), *Attention-Guided* (ours), and Δ_t*-Guided* (Ben-Kish et al., 2025; Ye et al., 2025). $\mathcal{W}$ denotes the last 32 tokens of the input prompt $\mathcal{P}$. Strategies are detailed in Sec. 3.

tokens (highlighted in the red box in Fig. 2 (a)):

$$\alpha_t = \sum_{i \in \mathcal{G}} \alpha_{i,t} = \sum_{i \in \mathcal{G}} C_i^\top \Big(\prod_{k=t+1}^{i} \bar{A}_k \Big) \bar{B}_t, \quad (5)$$

where $\mathcal{G}$ denotes the set of generated tokens. Under the Oracle setting shown in Fig. 2 (b) (red dashed curve), we retain tokens with the highest α_t and plot the attention retention ratio against various token retention budgets. Approximately 20% of the tokens account for 80% of the total attention, suggesting that an effective filtering scheme can retain most of the attention signal while discarding a significant portion of the tokens. Additional results in Appendix E confirm this sparsity pattern across different global channels.

3.2 Token Filtering Metric

In practice, token filtering (Ben-Kish et al., 2025; Ye et al., 2025) must be performed before generation begins to mitigate information decay in input prompts, necessitating proxy metrics for estimating future token importance. Prior studies have considered two approaches:

Δ_t-Guided Filtering. Prior works (Ben-Kish et al., 2025; Ye et al., 2025) estimate token importance based on the magnitude of Δ_t. The underlying rationale is that a large Δ_t increases $\bar{B}_t$ through Eq. 2, which subsequently amplifies α_t via Eq. 5.

Attention-Guided Filtering. For Transformer-based LLMs, SnapKV (Li et al., 2024) proposes using historical attention as a importance metric. We adapt this idea to SSMs: tokens that received

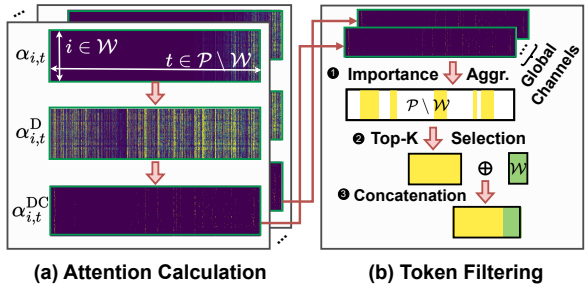


Figure 3: (a) The original Mamba attention $\alpha_{i,t}$ versus the proposed debiased attention $\alpha^{\mathrm{D}}_{i,t}$ and contrastive attention $\alpha^{\mathrm{DC}}_{i,t}$ for a 2048-token prompt. (b) End-to-end token filtering pipeline of LAMB.

substantial attention from the last few tokens before generation, referred to as the "observation window", are presumed influential for future generation. Formally, we define the importance metric as: $\hat{\alpha}_t = \sum_{i \in \mathcal{W}} \alpha_{i,t}$, where $\mathcal{W}$ denotes the observation window (last 32 tokens).

As shown in Fig. 2 (b), attention-guided filtering significantly outperforms Δ_t-guided filtering on the 512-token sequence, approaching the Oracle upper bound. At a 20% token retention ratio, attention-guided filtering retains about four times more attention in the generation window than Δ_t-guided filtering.

Despite its promise, attention-guided filtering presents two challenges in longer sequences:

C. 1 Bias Towards the Last Tokens: As illustrated in the first row of Fig. 3 (a), attention distributions in longer sequences show a strong bias towards the last few tokens (*i.e.,* the tokens closer to $\mathcal{G}$), with attention quickly diminishing for earlier tokens.

C. 2 Noisy Attention Patterns: The second row of Fig. 3 (a) highlights substantial noise in the attention distributions of longer sequences, making it difficult to identify truly influential tokens.

4 The Proposed LAMB Method

Motivated by the findings and challenges identified in Sec. 3, we introduce LAMB, a refined attention-guided token filtering framework designed to robustly handle long-context sequences in SSMs. We address the two challenges via enhanced attention metrics in Sec. 4.1, and then describe the end-to-end token filtering pipeline in Sec. 4.2.

4.1 Enhanced Attention Metric

Debiased Attention. To directly address **C. 1**, we introduce debiased attention, in which the cumulative decay factor $\prod_{k=t+1}^{i} \bar{A}_k$ responsible for the bias is replaced with a precomputed constant factor

$\prod_{k=1}^{L_0} \bar{A}_k$ based on the training sequence length length L_0, resulting in:

$$\alpha_{i,t}^{\mathrm{D}} = C_i^\top (\prod_{k=1}^{L_0} \bar{A}_k) \bar{B}_t. \tag{6}$$

As shown in Fig. 3 (a), this modification substantially reduces the bias towards recent tokens, yielding a more balanced representation of token importance.

Contrastive Attention. To address **C. 2**, we propose contrastive attention, which enhances the clarity of importance estimation by suppressing minor fluctuations and highlighting tokens with consistently strong contributions:

$$\alpha_{i,t}^{\mathrm{DC}} = \mathrm{ReLU}(\alpha_{i,t}^{\mathrm{D}} - \gamma \cdot \max_t(\alpha_{i,t}^{\mathrm{D}})), \tag{7}$$

where $\gamma < 1$ controls the strength of noise suppression, $\max_t \alpha_{i,t}^{\mathrm{D}}$ denotes the maximum attention score received by token i, and ReLU ensures non-negativity. The third row of Fig. 3 (a) clearly illustrates how contrastive attention effectively reduces noise and highlights the most influential tokens.

4.2 Aggregated Importance and Token Filtering Pipeline

While Sec. 4.1 introduces attention metrics for a single hidden state channel, an SSM typically comprises multiple channels. We extend the refined metrics across all global channels and present a three-step token filtering pipeline (illustrated in Fig. 3 (b)):

Step ❶ Importance Aggregation. For each token t, we aggregate the enhanced attention metric $\alpha_{i,t}^{\mathrm{DC}}$ across all global channels $c \in C_g$ and all tokens i in the observation window $\mathcal{W}$:

$$I_t^{\mathrm{raw}} = \sum_c \hat{\alpha}_{t,c}^{\mathrm{DC}} = \sum_c \sum_{i \in \mathcal{W}, i > t} \alpha_{i,t,c}^{\mathrm{DC}}. \tag{8}$$

We then apply average pooling to I_t^{raw} across the token dimension:

$$I_t = \texttt{mean\_pool}(I_t^{\mathrm{raw}}), \tag{9}$$

which preserves local context around the important tokens and improves long-context comprehension (see ablation results provided in Sec. 5.4).

Step ❷ Top-K Token Selection. We select the top-K tokens among $\mathcal{P} \setminus \mathcal{W}$ based on aggregated importance scores I_t, identifying those tokens most likely to influence future model generations.

Step ❸ Concatenation and Selective State Update. The selected influential tokens from $\mathcal{P} \setminus \mathcal{W}$ are concatenated with the tokens from the observation window $\mathcal{W}$. Hidden states are then updated exclusively based on this concatenated set, while the remaining tokens in the input prompt $\mathcal{P}$ are excluded from the state update. This selective update effectively reduces cumulative decay and preserves crucial context.

Together, these components enable LAMB to identify important tokens in SSMs and significantly improve long-context modeling.

5 Experiments

This section presents the evaluation results of LAMB: Sec. 5.3 and Sec. 5.2 detail the results on two long-context benchmarks, HELMET (Hsieh et al., 2024) and RULER (Yen et al., 2024); Sec. 5.4 presents an ablation study of the denoising and pooling techniques; finally, Sec. 5.5 measures the latency overhead of LAMB.

5.1 Experimental Setup

We apply the proposed LAMB method to a representative SSM Mamba-2 (Gu and Dao, 2024) and a hybrid model Zamba-2 (Glorioso et al., 2024b), and benchmark them against both vanilla models and LongMamba (Ye et al., 2025), the SOTA method to enhance the SSMs' long-context capabilities. Further setup details are in Appendix A.

5.2 Benchmark Results on HELMET

Tab. 1 presents the results on HELMET (Yen et al., 2024). The table shows that the proposed LAMB consistently achieves the highest average accuracy across all three context lengths (*i.e.,* 8k, 16k, 32k) on both Mamba2 and Zamba2. This consistent improvement highlights the effectiveness of LAMB at varying context lengths. The detailed per-task performance on HELMET is provided in Appendix C.

Table 1: Benchmark results on HELMET.

Model	Method	Sequence Length		
		8k	16k	32k
Mamba2	Vanilla	3.23	3.08	1.60
	LongMamba	7.27	6.31	6.20
	LAMB (Ours)	**10.63**	**10.25**	**7.82**
Zamba2	Vanilla	5.11	6.76	3.70
	LongMamba	12.21	11.35	9.17
	LAMB (Ours)	**13.93**	**12.35**	**11.28**

Table 2: Comparison of performance (%) on RULER with a 16k context length. We evaluate vanilla models, LongMamba, and LAMB on Mamba2 (Dao and Gu, 2024) and Zamba2 (Glorioso et al., 2024a).

Model	Method	single1	single2	single3	multi1	multi2	multi3	multivalue	multiquery	vt	cwe	fwe	qa1	qa2	Avg.
Mamba2-780M	Vanilla	0	0	0	0	0	0	0	0	0	0	0	0	2	0.15
	LongMamba	0	7	2	7	0	0	1.75	2.5	1.6	0	5	7	7	3.14
	LAMB (Ours)	**98**	**88**	**73**	**57**	**3**	0	**6**	**34**	**6**	**0.7**	**42.67**	**9**	**18**	**33.49**
Mamba2-1.3B	Vanilla	0	0	1	1	0	0	0.5	0	0	0	1	0	1	0.27
	LongMamba	**100**	2	2	5	0	0	0.75	0.75	1.4	0.4	2.33	8	18	10.82
	LAMB (Ours)	99	**90**	**78**	**40**	**6**	0	**6.5**	**24.5**	**14.6**	**0.5**	**47.33**	**14**	**21**	**33.96**
Zamba2-1.2B	Vanilla	30	11	7	6	0	0	8	0	0.2	0	13.67	0	1	5.92
	LongMamba	79	92	**31**	23	0	0	**58**	**49.25**	0.2	**2.3**	0.67	1	11	26.72
	LAMB (Ours)	**83**	**96**	**31**	**26**	**2**	**1**	49.75	43	**0.8**	0.5	**30.67**	**4**	**13**	**29.29**

5.3 Benchmark Results on RULER

Tab. 2 provides the results on RULER (Hsieh et al., 2024). It can be observed that the proposed LAMB provides a +30.35%/+23.14%/+2.57% performance improvement compared to LongMamba on Mamba2-780M/Mamba2-1.3B/Zamba2-1.2B. Notably, LAMB is consistently better than LongMamba on QA tasks (*i.e.,* qa1 and qa2), demonstrating the potential of LAMB on related real-world applications.

5.4 Ablation Study

Tab. 3 presents an ablation study of the proposed techniques on the RULER benchmark. Here, *Denoising* refers to whether the denoised attention $\alpha_{i,t}^{\mathrm{DC}}$ is used instead of the noisy $\alpha_{i,t}^{\mathrm{D}}$, and *Pooling* refers to whether mean pooling (Eq. 9) is applied to the per-token importance. The first row and second row of Tab. 3 demonstrate that *Pooling* is crucial for LAMB, as the average accuracy on RULER drops from 33.96% to less than 5% without it. The table shows that *Denoising* is also necessary to achieve a high accuracy with LAMB, which brings a 6.74% improvement in the average accuracy on RULER (comparing the last two rows of Tab. 3). Ablation studies on the denoising factor γ and pooling kernel size are presented in Appendix B.

5.5 Latency Overhead

Tab. 4 reports the prefill latency of LAMB across a range of prompt lengths. For context lengths ranging from 32k to 192k tokens, LAMB incurs at most 12.31% additional latency. The overhead falls below 7% for contexts of 96k tokens or more, highlighting the efficiency and scalability of LAMB.

Table 3: Ablation study of the proposed techniques on RULER with a 16k sequence length.

Model	Denoising	Pooling	Acc. (%)
Mamba2-1.4B	✗	✗	3.40
	✓	✗	4.52
	✗	✓	27.22
	✓	✓	**33.96**

Table 4: Prefill latency (in seconds) for the vanilla Mamba2-1.3B model (Dao and Gu, 2024) and its LAMB-enhanced variant on an NVIDIA A100 GPU (Choquette et al., 2021) (batch size = 1). LAMB adds *no* overhead during the generation stage.

Prompt Length	32k	64k	96k	128k	160k	192k
Vanilla	0.52	1.05	1.58	2.11	2.63	3.16
LAMB	0.58	1.14	1.69	2.24	2.79	3.34
Overhead (%)	12.31	8.06	6.90	6.21	5.84	5.78

6 Conclusion

We introduced LAMB, a training-free method for enhancing the long-context performance of SSMs and hybrid architectures by selectively filtering out tokens from the hidden state update. LAMB is the first attention-guided method for token filtering and achieves up to an average of 30.35% improvement over the state-of-the-art (SOTA) method. We anticipate our findings to spark broader applications of token filtering in long-sequence modeling.

7 Limitations

LAMB presently relies on fixed values for the denoising factor γ and the pooling kernel size. Such rigidity can constrain its effectiveness across varied tasks. Enabling these hyperparameters to adjust dynamically to the input and the long-context task might unlock additional gains, which we leave to future work.

Acknowledgment

This work was partially supported by CoCoSys, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, and National Science Foundation (NSF) Division of Information & Intelligent Systems (IIS) program (Award ID: 2403297).

References

Ameen Ali, Itamar Zimerman, and Lior Wolf. 2024. The hidden attention of mamba models. *arXiv preprint arXiv:2403.01590*.

Seyedarmin Azizi, Souvik Kundu, Mohammad Erfan Sadeghi, and Massoud Pedram. 2025. Mambaextend: A training-free approach to improve long context extension of mamba. In *The Thirteenth International Conference on Learning Representations*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Assaf Ben-Kish, Itamar Zimerman, Shady Abu-Hussein, Nadav Cohen, Amir Globerson, Lior Wolf, and Raja Giryes. 2025. Decimamba: Exploring the length extrapolation potential of mamba. In *The Thirteenth International Conference on Learning Representations*.

Jack Choquette, Wishwesh Gandhi, Olivier Giroux, Nick Stam, and Ronny Krashinsky. 2021. Nvidia a100 tensor core gpu: Performance and innovation. *IEEE Micro*, 41(2):29–35.

Tri Dao and Albert Gu. 2024. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *Preprint*, arXiv:2405.21060.

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, Zijia Chen, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van Keirsbilck, Min-Hung Chen, Yoshi Suhara, Yingyan Lin, Jan Kautz, and Pavlo Molchanov. 2024. Hymba: A hybrid-head architecture for small language models.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, Anna Golubeva, Vasudev Shyam, James Whittington, Jonathan Pilault, and Beren Millidge. 2024a. The zamba2 suite: Technical report. *arXiv preprint arXiv:2411.15242*.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. 2024b. Zamba: A compact 7b ssm hybrid model. *Preprint*, arXiv:2405.16712.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. *Preprint*, arXiv:2312.00752.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. 2022. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35:35971–35983.

Albert Gu, Karan Goel, and Christopher Ré. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

R. E. Kalman. 1960. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. Snapkv: Llm knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, volume 37, pages 22947–22970. Curran Associates, Inc.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. 2024. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*.

Llama Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. 2023. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pages 26670–26698. PMLR.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. 2024. Samba: Simple hybrid state space models for efficient unlimited context language modeling. *arXiv preprint*.

Chien Van Nguyen, Huy Huu Nguyen, Thang M Pham, Ruiyi Zhang, Hanieh Deilamsalehy, Puneet Mathur, Ryan A Rossi, Trung Bui, Viet Dac Lai, Franck Dernoncourt, et al. 2024. Taipan: Efficient and expressive state space language models with selective attention. *arXiv preprint arXiv:2410.18572*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, Garvit Kulshreshtha, Vartika Singh, Jared Casper, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. 2024. An empirical study of mamba-based language models. *Preprint*, arXiv:2406.07887.

Zhifan Ye, Kejing Xia, Yonggan Fu, Xin Dong, Jihoon Hong, Xiangchi Yuan, Shizhe Diao, Jan Kautz, Pavlo Molchanov, and Yingyan Celine Lin. 2025. Longmamba: Enhancing mamba's long-context capabilities via training-free receptive field enlargement. In

The Thirteenth International Conference on Learning Representations.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2024. Helmet: How to evaluate long-context language models effectively and thoroughly. *Preprint*, arXiv:2410.02694.

A Experimental Setup

Models and Benchmarks. We evaluate LAMB with two open-sourced SSMs (*i.e.,* Mamba2-780M, Mamba2-1.4B (Gu and Dao, 2024)) and a representative hybrid model Zamba2-1.3B (Glorioso et al., 2024b) on two long-context benchmarks: HELMET (Yen et al., 2024) and RULER (Hsieh et al., 2024). Specifically, HELMET encompasses a diverse range of application-centric long-context tasks, covering synthetic recall, long-document question answering, summarization, many-shot in-context learning, retrieval-augmented generation, passage re-ranking, and citation-aware generation. Unlike HELMET, RULER consists exclusively of synthetic tasks, allowing precise control over sequence length and task complexity. Comprehensive evaluation on both benchmarks provides a robust assessment of LAMB's ability to handle diverse long-context applications.

Baselines. We benchmark our proposed LAMB against two baselines: vanilla models without token filtering and LongMamba (Ye et al., 2025), the SOTA training-free method for enhancing Mamba models' long context understanding via token filtering.

Hyperparameters. For all experiments, we use a denoising factor γ=0.9. When testing Mamba2 models on RULER, the pooling kernel size is 51 and we preserve top 512 tokens in each Mamba block. On all other experiments, we preserve top 1024 tokens in each Mamba block and set the pooling kernel size to 9 for the Zamba-1.2B model and 18 for the Mamba2 models. We use the same hyperparameters in LongMamba (Ye et al., 2025) for determining the global channels.

Evaluation Protocol. Our evaluation follows the official implementation and settings of the benchmarks. More specifically, on the RULER benchmark, we generate 100 test sequences for each task under each sequence length. The metrics used for HELMET and RULER follow their original papers (Yen et al., 2024; Hsieh et al., 2024), respectively.

B Ablation Study on Hyperparameters

B.1 Ablation study on the Denoising Factor

Tab. 5 shows the impact of varying the denoising factor γ on LAMB's performance across tasks in the RULER benchmark. As γ increases from 0 to 0.9, we observe a consistent improvement in most metrics. We also find that the average performance differs by less than 2% for γ values between 0.8 and 0.95, indicating that LAMB is robust to the exact choice of γ in this range.

B.2 Ablation study on Pooling Kernel Size

Tab. 6 presents the performance impact of different pooling kernel sizes on the RULER benchmark. Without pooling (kernel size = 1), performance is notably poor. Increasing the kernel size to 51 yields steady improvements, whereas using a size of 71 causes a drop in average accuracy. These results suggest that (1) pooling with a sufficiently large kernel is crucial for enhancing LAMB's performance, and (2) an oversized kernel can degrade effectiveness.

C Detailed Experimental Results

Tab. 7 presents the detailed per-task performance on HELMET. We evaluated the vanilla models, LongMamba, and HELMET using both Zamba2-1.2B and Mamba2-1.3B.

D Extended Related Works

State Space Models. The quadratic space and time complexity of attention mechanisms (Vaswani, 2017) has posed significant challenges in training and inference with long input sequences. To address these issues, a new class of models that replaces attention with convolution-based architectures has recently gained traction, offering a more computationally feasible alternative (Gu et al., 2021). Among these alternatives, state space models (SSMs) have emerged as a powerful framework. SSMs, extensively used in control theory, describe the evolution of dynamic systems through latent states that encapsulate system behavior over time (Kalman, 1960). Several models, including S4 (Gu et al., 2021), S4D (Gu et al., 2022), and LRU (Orvieto et al., 2023), leverage this approach. Notably, Mamba models (Gu and Dao, 2024; Dao and Gu, 2024) have gained attention for its departure from traditional time-invariant formulations by introducing a time-dependent state update mechanism, thereby enhancing expressivity and adaptability in sequence modeling tasks.

Long Context Mamba. While Mamba models have demonstrated strong performance under short context, prior studies (Waleffe et al., 2024; Ben-Kish et al., 2025) have shown that their effectiveness diminishes on long-context tasks due to exponential hidden state decay, which limits their ability to retain information over extended sequences. To

Table 5: LAMB's performance (%) on RULER (16k context length) with different γ values.

γ	single1	single2	single3	multi1	multi2	multi3	multivalue	multiquery	vt	cwe	fwe	qa1	qa2	Avg.
0	95	72	34	25	0	0	5.50	11.50	28.4	0.2	47.33	15	20	27.23
0.5	91	89	78	33	8	0	6.75	12.00	17.2	0.4	31.33	12	16	30.36
0.8	97	94	77	37	6	0	7.25	25.50	10.4	0.6	39.00	10	21	32.67
0.9	99	90	78	40	6	0	6.50	24.50	14.6	0.5	47.33	14	21	33.96
0.95	100	91	74	41	2	0	7.75	27.25	15.8	0.5	44.33	11	21	33.51

Table 6: LAMB's performance (%) on RULER (16k context length) with different pooling kernel sizes.

Kernel	single1	single2	single3	multi1	multi2	multi3	multivalue	multiquery	vt	cwe	fwe	qa1	qa2	Avg.
1	2	2	1	0	0	0	0.50	0.50	4.2	0.6	18.00	11	19	4.52
31	99	96	10	40	3	0	10.00	34.75	8.4	0.3	38.67	13	26	29.16
51	99	90	78	40	6	0	6.50	24.50	14.6	0.5	47.33	14	21	33.96
71	100	91	82	43	5	0	7.00	28.25	16.0	0.5	42.00	8	18	33.90

address this limitation, several methods have been proposed. For instance, DeciMamba (Ben-Kish et al., 2025) mitigates hidden state decay by progressively filtering out tokens across layers based on the magnitude of Δ_t. LongMamba (Ye et al., 2025), on the other hand, introduces the concept of global channels—channels with a wide receptive field that capture long-range dependencies across tokens—and selectively applies token filtering to these channels. Despite these advancements, the underlying principles behind the effectiveness of token filtering and its potential for further optimization remain unclear, which is a key focus of this study.

Hybrid Transformer-SSM Models. To complement the long-range precision of self-attention with the efficiency of SSM blocks, recent work has proposed hybrid architectures that employs both in a single network. For instance, Hymba (Dong et al., 2024) mixes attention and SSM heads in parallel within each layer, while a second line of work, exemplified by Jamba (Lieber et al., 2024), Samba (Ren et al., 2024), and Zamba (Glorioso et al., 2024b,a), alternates attention and Mamba layers throughout the network depth. Both approaches achieve compelling accuracy–efficiency trade-offs compared with purely Transformer- or Mamba-based models. In parallel, Taipan (Van Nguyen et al., 2024) augments Mamba with a lightweight selective-attention module that focuses on long-range dependencies among salient tokens. Orthogonal to these efforts, our proposed LAMB serves as a plug-and-play enhancement for pretrained SSM or hybrid models, boosting long-context performance without further training.

E Additional Visualization

Visualizations of attention maps and the attention retention ratios under different approaches for additional global channels are provided in Fig.4 and Fig.5. In this section, we follow the same experimental setups as in Fig. 2 and randomly sample eight additional global channels from the Mamba2-1.3B model for visualization.

Table 7: Detailed benchmark results on HELMET.

	Methods	Recall			RAG			Cite			Re-rank			LongQA			Summ			ICL			Avg		
		8k	16k	32k	8k	16k	32k	8k	16k	32k	8k	16k	32k	8k	16k	32k	8k	16k	32k	8k	16k	32k	8k	16k	32k
Mamba2	vanilla	0	0	0	11.9	10.2	5.2	0.0	0.1	0.0	0.2	0.0	0.0	5.3	4.2	3.2	1.9	1.3	0.8	11.8	5.8	2.1	3.2	3.1	1.6
	LongMamba	2.1	1.2	3.0	13.1	14.3	15.4	0.0	0.3	0.2	3.9	1.3	0.0	15.3	16.2	15.4	6.3	4.6	4.3	10.2	6.3	5.1	7.3	6.3	6.2
	LAMB	4.1	5.4	4.2	25.6	24.9	20.8	1.4	0.6	0.7	0.1	0	0	11.7	11.1	11.1	7.2	6.4	6.2	24.2	23.4	11.7	**10.6**	**10.3**	**7.8**
Zamba2	vanilla	0.3	1.2	0.0	12.7	14.2	4.9	0.6	1.0	0.3	0.1	0.7	0.5	4.7	4.6	5.0	3.6	2.1	4.4	13.8	23.6	10.8	5.1	6.8	3.7
	LongMamba	13.8	7.9	4.1	29.6	28.8	22.3	0.4	0.8	1.0	9.2	4.2	0.7	5.7	6.6	7.0	4.7	5.5	1.8	22.0	25.6	27.2	12.2	11.4	9.2
	LAMB	12.4	6.8	5.5	26.3	24.0	20.0	1.1	1.1	0.7	16.0	2.3	3.8	10.6	10.5	9.3	7.1	6.8	5.8	24.0	35.2	33.2	**13.9**	**12.4**	**11.3**

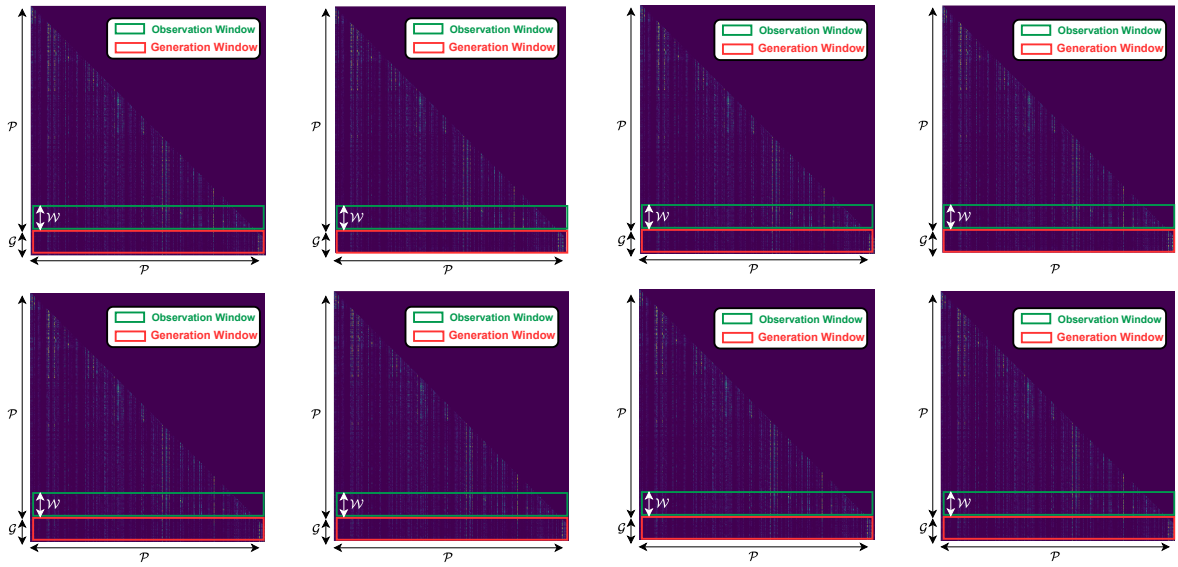


Figure 4: Visualization of attention maps from eight randomly sampled global channels, covering 512 input tokens (denoted as $\mathcal{P}$) and 32 generated tokens (denoted as $\mathcal{G}$). $\mathcal{W}$ denotes the last 32 tokens of $\mathcal{P}$.

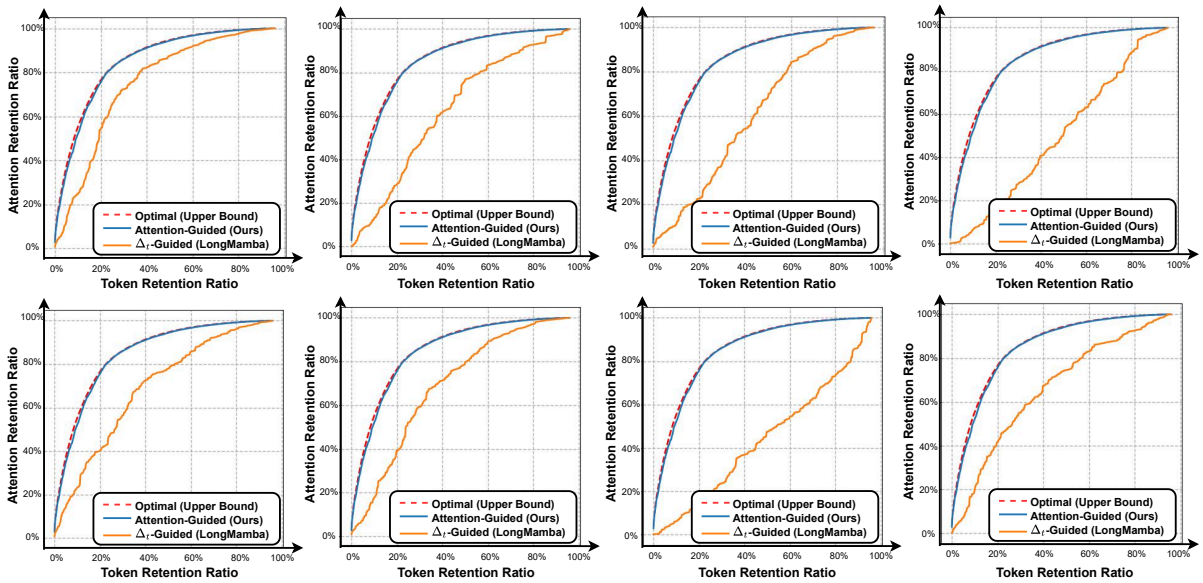


Figure 5: Attention retention ratio within the generation window under varying token retention budgets. Here we evaluate three token filtering strategies: *Oracle* (upper bound, requires oracle access to future attention maps), *Attention-Guided* (ours), and Δ_t*-Guided* (Ben-Kish et al., 2025; Ye et al., 2025).

Counterfactual-Consistency Prompting for Relative Temporal Understanding in Large Language Models

Jongho Kim and Seung-won Hwang*
Interdisciplinary Program in Artificial Intelligence, Seoul National University
{jongh97, seungwonh}@snu.ac.kr

Abstract

Despite the advanced capabilities of large language models (LLMs), their temporal reasoning ability remains underdeveloped. Prior works have highlighted this limitation, particularly in maintaining temporal consistency when understanding events. For example, models often confuse mutually exclusive temporal relations like "before" and "after" between events and make inconsistent predictions. In this work, we tackle the issue of temporal inconsistency in LLMs by proposing a novel counterfactual prompting approach. Our method generates counterfactual questions and enforces collective constraints, enhancing the model's consistency. We evaluate our method on multiple datasets, demonstrating significant improvements in event ordering for explicit and implicit events and temporal commonsense understanding by effectively addressing temporal inconsistencies.

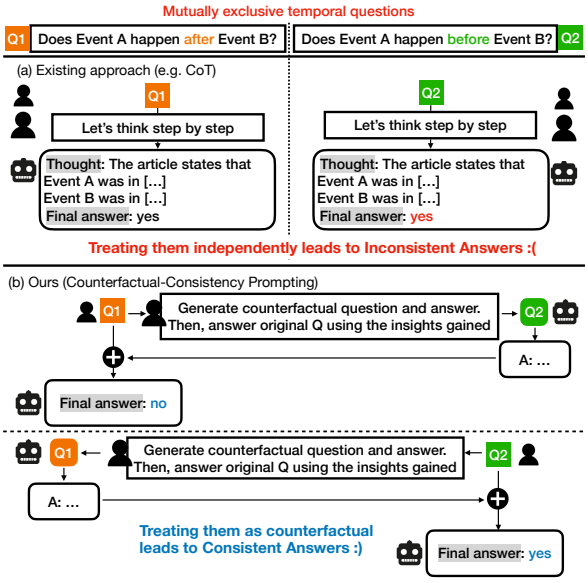


Figure 1: Example of leveraging counterfactual questions to resolve temporal inconsistency in LLMs.

1 Introduction

Despite the impressive capabilities of LLMs, a line of research (Jain et al., 2023; Chu et al., 2023) has highlighted that these models often lack temporal reasoning abilities. This is especially true for *relative* event understanding, where the goal is to infer temporal relationships between events or properties within an event in the passage, without depending on *absolute* time indicators (e.g., specific dates).

The primary challenge is that LLMs lack *temporal consistency* in their responses (Qiu et al., 2024; Chen et al., 2024). Temporal consistency is defined as the model's ability to ensure that conflicting timelines do not co-exist. For instance in Figure 1-(a), if the model is temporally inconsistent, mutually exclusive temporal relations like "before" and "after" are sometimes confused when ordering events, leading to contradictory predictions—such as stating that Event A happens both before and after Event B in the same context.

While events with time indicators are often addressed with mathematical reasoning (Zhu et al., 2023; Su et al., 2024), no existing work has successfully tackled the challenge of temporal inconsistency in the events' relative understanding without requiring explicit time markers. Chain-of-thought (CoT) reasoning (Wei et al., 2022), which primarily aids mathematical and symbolic reasoning (Sprague et al., 2024), is also reported to fail to solve such inconsistency (Qiu et al., 2024). Considering temporal consistency is fundamental in temporal reasoning, its absence in LLM can undermine key tasks like planning (Sakaguchi et al., 2021; Zhang et al., 2024). These observations highlight the need for alternative reasoning skills to achieve temporal consistency.

This study answers the following research question: **Can we prompt LLMs to elicit the ability to mitigate temporal inconsistency?** Inspired by counterfactual augmentation, where models are exposed with lexically similar, but typically label-

*Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1210–1225
July 27 - August 1, 2025

flipping pairs in training (Kaushik et al., 2020), we extend it to LLMs to generate *temporally counterfactual questions*: We introduce lexically small interventions to the original input (e.g. before to after, years to centuries) that drastically affect its temporal semantics. By providing these questions and self-generated answers alongside the original input, the model would rely less on lexical similarities and better understand the semantics.

To this end, we propose a novel counterfactual-consistency prompting (CCP), designed to enhance the temporal consistency of LLMs, as described in Figure 1-(b). CCP first generates temporal counterfactual exemplars and then applies the insights gained to address the original temporal question. This method is particularly effective in relative event understanding because the counterfactual exemplars not only encourage the model to understand different temporal semantics but also directly impose temporal constraints. For instance, if the model states that "Event A happens after Event B" and also recognizes that "Event A happens before Event B", the conflict forces the model to collectively re-weight the validity of these two statements.

We show performance gain of CCP across multiple relative event understanding tasks. Our effectiveness in mitigating temporal inconsistencies is further demonstrated by our inconsistency metric.

2 Method

Our idea is to make the model approximate the *temporal constraints* using counterfactuals. Because temporal reasoning imposes unique interdependencies, where one temporal aspect affects another (Han et al., 2019; Kim et al., 2024), counterfactuals enable us to capture these constraints. For example, if the model establishes from a counterfactual exemplar that "Event e_1 happens [r_2: before] Event e_2", it is constrained to predict the original question that "Event e_1 cannot happen [r_1: after] Event e_2":

$$r_2(e_1, e_2) \in \mathcal{V} \implies r_1(e_1, e_2) \notin \mathcal{V} \quad (1)$$

where $r(e_a, e_b)$ represents the temporal relation r between events e_a and e_b, and $\mathcal{V}$ represents the set of coherent temporal relations with the context.

2.1 Generating Temporally Counterfactual Questions

Given a context C, our task is to provide an accurate answer to question Q while maintaining temporal consistency. We start by creating *temporally counterfactual* questions, $Q^{c_1}...Q^{c_i}$.

Counterfactual augmentation conventionally aims to generate instances with lexically minimal edits while keeping others unchanged (Huang et al., 2019; Kaushik et al., 2020; Wang and Culotta, 2020). This strategy discourages models from relying too much on superficial similarity. However, previous works (Kaushik et al., 2020) require arbitrary label-flipping edits, which are unsuitable for LLM inference with unknown test labels.

In contrast, we focus on *temporally counterfactual* questions that specifically edit the temporal semantics of the original sentence. Our approach ensures that the model adheres to the "temporal constraints", yet retains the effect of label-flipping that emphasizes temporal cues over superficial similarity.

Specifically, we set the types of counterfactual questions based on the temporal semantics each dataset aims to capture. The types of temporal counterfactuals are listed in Table 1. For sentences representing temporal relation between two events ($r_1(e_1, e_2)$), we substitute the relation r_1 to r_2, or event e_2 to e_3. For those representing an event's temporal properties ($r_1(e_1)$) such as duration or stationarity, we substitute the property to r_2 or negate it to $\neg r_1$.

To generate counterfactual questions, we design our model to dynamically create them rather than rely on a predefined rule-based template. While rule-based approaches like Chen et al. (2024) in logical reasoning constrain answers using a predefined question set, they limit flexibility to cover the broad range of temporal expressions. By comparison, our dynamic generation of constraints provides a more adaptable solution. We specify various counterfactual types through in-context learning (ICL) to control the relevance of these generated questions. The full prompts are in Appendix F.1.

2.2 Counterfactual-Consistency Prompting

After generating the counterfactual questions, we prompt the model again to produce predictions for counterfactual $Y^{c_1}, ..., Y^{c_n}$. However, there is a risk when LLMs may fail to answer the counterfactual questions correctly. In this case, their direct use propagates errors to the original question.

As a proxy for determining whether the generated prediction can be trusted, existing works aggregate multiple predictions of the same ques-

Dataset	Relation	Revision Type	Example
TempEvalBi-QA	$r_1(e_1, e_2)$	Original	Did they get married **after** they moved to Maine?
	$r_2(e_1, e_2)$	$r_1 \rightarrow r_2$	Did they get married **before** they moved to Maine?
TRACIE	$r_1(e_1, e_2)$	Original	Chad found the cap **before** he got off of the ride.
	$r_2(e_1, e_2)$	$r_1 \rightarrow r_2$	Chad found the cap **after** he got off of the ride.
MCTACO: Duration	$r_1(e_1)$	Original	It has existed for **1 year**.
	$r_2(e_1)$	$r_1 \rightarrow r_2$	It has existed for **centuries**.
MCTACO: Frequency	$r_1(e_1)$	Original	It rained **a couple of times** in the summer.
	$r_2(e_1)$	$r_1 \rightarrow r_2$	It rained **every month** in the summer.
MCTACO: Stationarity	$r_1(e_1)$	Original	She is **still** in Ranchipur.
	$r_2(e_1)$	$r_1 \rightarrow r_2$	She is **not** in Ranchipur.
MCTACO: Ordering	$r_1(e_1, e_2)$	Original	They **went to the store** after they were put in jail.
	$r_1(e_1, e_3)$	$e_2 \rightarrow e_3$	They **repented** after they were put in jail.
MCTACO: Typical Time	$r_1(e_1)$	Original	Durer died **40 years** later.
	$r_2(e_1)$	$r_1 \rightarrow r_2$	Durer died **360 years** later.

Table 1: The temporal aspects targeted by each dataset and types of counterfactuals addressed by our framework. The examples illustrate how counterfactual questions modify the semantics regarding temporal relations (r_1,r_2) for events (e_1,e_2,e_3).

tion (Wang et al., 2023; Du et al., 2024). Formally, the refined prediction Y is derived by re-weighting the probability distribution P of previous predictions $Y_1, ..., Y_n$ from the same question as: $P(Y) = f(P(Y_1), ..., P(Y_n))$ where f is an aggregation function such as majority voting or LLM itself. Though, they can lead to errors as they solely rely on feedback from a single question.

Our distinction is to aggregate predictions from both the original and counterfactual questions. We design the model to re-weight the counterfactual answer distributions across the questions.

$$P(Y) = f(P(Q, Y), P(Q^{c_1}, Y^{c_1}), ..., P(Q^{c_n}, Y^{c_n})) \quad (2)$$

For instance, even if the model wrongly predicts the relation as 'after' in a counterfactual, collectively considering the possibility of the relation 'before' can re-weight the effect of the constraint. The prompts are provided in Appendix F.2.

This re-evaluation approach improves robustness against potential errors in generated answers. The second analysis in Subsection 3.5 shows such self-correction outperforms a baseline directly leveraging counterfactuals without aggregation.

3 Experiments

3.1 Datasets

Among publicly available datasets, we selected three based on two criteria: (1) the task focuses on relative event understanding without absolute time indicators, and (2) the temporal inconsistency on the dataset can be evaluated.

TempEvalQA-Bi (Qiu et al., 2024) involves ordering two explicit events in time, assessing temporal consistency in mutually exclusive question pairs. **TRACIE** (Zhou et al., 2021) expands the event ordering to implicit events, testing if the hypothesis logically follows the story. We finally added **MCTACO** (Zhou et al., 2019) considering the diverse event-related temporal properties. The dataset covers broader aspects like event duration or frequency. We modified the multiple-choice setting of MCTACO into a binary question-answering task for consistency evaluation, presenting each answer candidate separately to determine if it fits the context. Dataset statistics and examples are in Appendix A.

3.2 Metrics

Along with accuracy (ACC) and F1 scores to assess overall performance, we introduce the inconsistency metric (INC) as a main evaluation measure for temporal inconsistency. We define the INC as the percentage of inconsistent predictions. An inconsistency is counted when at least one incorrect answer is found within a group of minimally dissimilar questions with slight modifications in their temporal semantics, while all other aspects remain unchanged.

TempEvalQA-Bi directly provides this metric. For TRACIE, we manually group questions that are counterfactual to each other. We adapt INC in MCTACO by grouping original multiple-choice candidates by question.

3.3 Evaluation Settings and Baselines

For models, we used open-source LLM **Llama-3 8B** and **70B** (AI@Meta, 2024), and API-based LLM **GPT-4o-mini** and **GPT-4o** (OpenAI et al.,

		TempevalQA-Bi			TRACIE			MCTACO		
		ACC	F1	INC (↓)	ACC	F1	INC (↓)	ACC	F1	INC (↓)
Llama -3-8B	SP	65.4	63	57.6	57.4	66.9	75.2	77.7	69.4	59.8
	CoT	69.6	70.6	50	63	64.9	56	77.6	69.8	63.4
	Consistency	70.8	71.2	49.6	64.9	67.3	57.8	77.5	69.0	61.1
	Reflection	63.6	63.9	44.6	62.5	55.7	55.5	77.4	69.7	76.4
	Debate	67.6	65.2	52.2	63.6	66	53.2	37.4	31.6	88.1
	CCP	**75.9**	**75.2**	**32.7**	**68.8**	**70.4**	**39.8**	**78.7**	**81.7**	**57.7**
Llama -3-70B	SP	76.6	78.6	39.7	79.9	79.7	29.6	85.2	81.8	43.5
	CoT	80.4	82	31.3	80.1	80	31.8	85.9	82.2	46.9
	Consistency	-	-	-	-	-	-	-	-	-
	Reflection	77	77.9	35.3	80	78.3	30.3	80.6	73	56.5
	Debate	81	82.8	32.6	81.6	80.7	25.9	85.3	81.4	45.9
	CCP	**87.3**	**87.9**	**19.2**	**86.5**	**86.1**	**12.0**	**89.4**	**87.6**	**37.0**
GPT-4o -mini	SP	78.8	76.4	36.6	74.6	71.3	38.2	76.0	63.1	65.8
	CoT	81.3	79.9	29	73.2	68.5	42.7	80.9	73.7	58.9
	Consistency	85.5	85.5	21.9	73.6	68.8	42.8	78.9	69.4	60.6
	Reflection	86.8	86.9	22.8	74.4	70.9	39.1	74.8	60.2	68.5
	Debate	86.4	86.4	24.6	73	67.1	44.5	78.3	68.2	61.0
	CCP	**88.8**	**88.7**	**19.6**	**82.5**	**81.2**	**20.2**	**87.8**	**85.8**	**42.1**
GPT-4o	SP	86.4	85.8	20.1	80.1	78.6	27.0	79.7	70.9	60.5
	CoT	90.4	90	17.4	80.2	78.1	32.4	84.4	80	49.7
	Consistency	91.7	91.5	14.7	80.1	77.7	31.4	82.9	77.3	49.7
	Reflection	93.1	93	11.2	82.7	80.9	26.6	80.0	72.2	55.4
	Debate	90.8	90.6	11.2	80.6	77.9	32.8	81.4	74.6	52.2
	CCP	**93.8**	**93.8**	**8.0**	**85.8**	**84.7**	**17.6**	**90.4**	**88.8**	**35.0**

Table 2: The full performance comparison results on the relative event understanding tasks. Our prompting methods, which leverage self-generated exemplars as the temporal constraint, outperform baselines across the board.

2024).

For baselines, we first compare CCP with standard prompting (**SP**) that directly answers the question without intermediate steps, and **CoT**, which incorporates step-by-step reasoning to derive the answer. Next, we consider methods that aggregate multiple predictions of the same question. Self-**Consistency** (Wang et al., 2023) predicts one question multiple times and performs majority voting. Self-**Reflect** methods (Madaan et al., 2024; Shinn et al., 2024) iteratively refine own predictions. Multi-agent **Debate** (Du et al., 2024) leverages both majority vote and reflection. We employ a 3-shot setting across all configurations. More details on evaluation settings are in Appendix B.

3.4 Main results

Table 2 highlights the performance of our method compared to baseline methods on relative event understanding tasks. Compared to SP, the CoT baseline is not usually effective and often worsens performance. Advanced baselines, Consistency, Reflect, and Debate, also fail to consistently reduce inconsistencies or achieve competitive accuracy. In contrast, CCP steadily outperforms these baselines across various LLMs, significantly reducing temporal inconsistencies across all datasets and achieving notable improvements in ACC and F1 scores.

3.5 Analysis

Creating counterfactual questions by generation handles diverse temporal aspects. We compared our generative setting with the **Ret**rieved **Q**uestions (Ret.Q) approach, where counterfactual questions were retrieved from other questions within the same question group. We evaluated the methods on MCTACO, which covers various aspects of event reasoning.

Figure 2 shows that generating counterfactual questions proved more effective for all temporal

types. These results suggest that our method performs better in event understanding with diverse relations, where the dataset cannot often provide high-quality counterfactual questions. Notably, CCP outperforms the Ret.Q baseline even though our method may produce incorrect questions. Also, CCP is more practical since Ret.Q assumes the questions in the test set are observed.

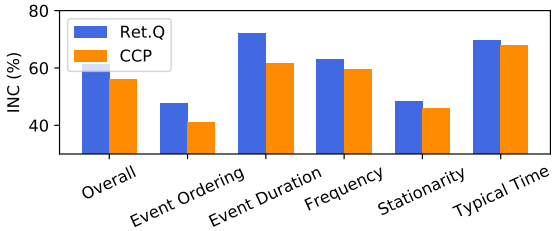


Figure 2: Comparison between counterfactual example collection methods on MCTACO with Llama-3-8B.

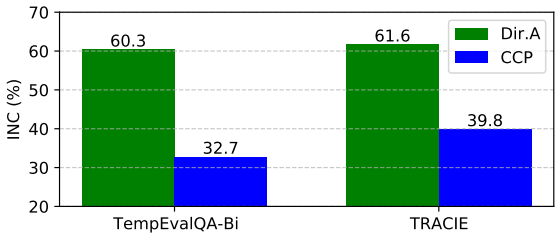


Figure 3: Comparison between different counterfactual leveraging methods with the Llama-3-8B model.

CCP is robust against wrong counterfactual exemplars. We conducted a comparative analysis of two methods: **Dir**ect **A**nswering (Dir.A), which involves answering directly from counterfactual exemplars, versus CCP which leverages the aggregation step to re-evaluate them. We conducted experiments on TempEvalQA-Bi and TRACIE, where before-after relations ensure that identifying a counterfactual answer is sufficient to determine the original. We excluded MCTACO since its counterfactual answers do not always determine the validity of the original answer. In the Dir.A implementation, the answer to the counterfactual question is flipped and directly used as the response to the original question. The results in Figure 3 demonstrate that CCP consistently outperforms Dir.A, supporting our robustness by the collective evaluation.

CCP's performance gain is orthogonal to the number of ICL examples. Our approach inevitably introduces additional counterfactual examples during in-context learning (ICL), leading to a higher total number of shots compared to the baseline. To ensure a more competitive baseline, we increased the total number of shots in the baseline. In the MCTACO dataset and with the Llama model, we additionally experimented with the 12-shot CoT, which includes 12 passage (P)-question (Q)-candidate (C) pairs, and compared them with our 3-shot. We note that our 3-shot examples include 3 passage-question pairs and 11 candidates.

The results in Table 3 demonstrate that our method significantly outperforms the CoT, even with the increased number of examples in the baseline (INC score: 60.0 for CoT vs. 57.7 for Ours). This indicates that the performance gains are not simply due to the inclusion of more examples but are primarily driven by leveraging temporal constraints through counterfactual questions to enhance reasoning.

Additionally, we tested whether our approach benefits from additional ICL examples. The results in the last row of Table 3 confirm this, showing an improvement in INC score from 57.7 to 49.8, further validating the potential performance gain of our method.

	MCTACO				
	#P-Q	#C	ACC	F1	INC (↓)
CoT	3	3	77.6	69.8	63.4
CoT	12	12	78.9	72.2	60.0
CCP	3	11	78.7	81.7	57.7
CCP	12	26	**85.0**	**82.2**	**49.8**

Table 3: Performance comparison of Llama-3-8B on MCTACO with the different number of ICL examples.

4 Conclusion

We targeted the temporal inconsistency in relative event understanding with LLMs by proposing a prompting approach using counterfactual questions. This encourages the model to focus more on the temporal aspects and collectively evaluate its answer with imposed constraints. Experiments with the INC metric show that our approach mitigates inconsistency and improves overall performance.

5 Limitation

Our method showed limited performance improvement when time indicators, such as specific years

(e.g., 1980), are involved in temporal understanding. This is implied from our evaluations on event-time ordering and time-time ordering tasks, as shown in Appendix D.1. The findings suggest that arithmetic reasoning is essential for grounding timelines with absolute time indicators, as emphasized in prior studies (Su et al., 2024; Zhu et al., 2023).

Another limitation is that we focused on pointwise and pairwise event reasoning to highlight the model's struggles with basic temporal reasoning due to consistency issues. We anticipate future work expanding our approach to more complex listwise ordering like event schema prediction (Zhang et al., 2024).

Finally, the effectiveness of LLMs can be impacted by the specific wording of prompts (Jiang et al., 2020). For example, Appendix D.2 shows that, although our revised prompts generally yield better results than CoT, CCP experienced a slight decrease in performance.

Acknowledgement

This work was supported by Naver, the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2024-00414981), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00077/RS-2022-II220077, AI Technology Development for Commonsense Extraction, Reasoning, and Inference from Heterogeneous Data), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)].

References

AI@Meta. 2024. Llama 3 model card.

Meiqi Chen, Yubo Ma, Kaitao Song, Yixin Cao, Yan Zhang, and Dongsheng Li. 2024. Improving large language models in event relation logical prediction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9451–9478.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2023. Timebench: A comprehensive evaluation of temporal reasoning abilities in large language models. *arXiv preprint arXiv:2311.17667*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*.

Yi Fang, Moxin Li, Wenjie Wang, Hui Lin, and Fuli Feng. 2024. Counterfactual debating with preset stances for hallucination elimination of llms.

Rujun Han, I-Hung Hsu, Mu Yang, Aram Galstyan, Ralph Weischedel, and Nanyun Peng. 2019. Deep structured neural network for event temporal relation extraction. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 666–106.

Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. 2019. Reducing sentiment bias in language models via counterfactual evaluation. *arXiv preprint arXiv:1911.03064*.

Raghav Jain, Daivik Sojitra, Arkadeep Acharya, Sriparna Saha, Adam Jatowt, and Sandipan Dandapat. 2023. Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6750–6774, Singapore. Association for Computational Linguistics.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Jongho Kim, Dohyeon Lee, Minsoo Kim, and Seungwon Hwang. 2024. Chaining event spans for temporal relation grounding. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1689–1700, St. Julian's, Malta. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo Ponti, and Shay Cohen. 2024. Are large language model temporally grounded? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7064–7083, Mexico City, Mexico. Association for Computational Linguistics.

Keisuke Sakaguchi, Chandra Bhagavatula, Ronan Le Bras, Niket Tandon, Peter Clark, and Yejin Choi. 2021. proScript: Partially ordered scripts generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2138–2149, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183*.

Romain Storaï and Seung-won Hwang. 2024. Harp: Hesitation-aware reframing in transformer inference pass.

Zhaochen Su, Jun Zhang, Tong Zhu, Xiaoye Qu, Juntao Li, Min Zhang, and Yu Cheng. 2024. Timo: Towards better temporal reasoning for language models. *arXiv preprint arXiv:2406.14192*.

Shivin Thukral, Kunal Kukreja, and Christian Kavouras. 2021. Probing language models for understanding of temporal expressions. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 396–406.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Zhao Wang and Aron Culotta. 2020. Identifying spurious correlations for robust text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3431–3440.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Liang Yao. 2024. Large language models are contrastive reasoners.

Xinliang Frederick Zhang, Nicholas Beauchamp, and Lu Wang. 2024. Narrative-of-thought: Improving temporal reasoning of large language models via recounted narratives. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16507–16530, Miami, Florida, USA. Association for Computational Linguistics.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369.

Ben Zhou, Kyle Richardson, Qiang Ning, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2021. Temporal reasoning on implicit events from distant supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1361–1371.

Xinyu Zhu, Cheng Yang, Bei Chen, Siheng Li, Jian-Guang Lou, and Yujiu Yang. 2023. Question answering as programming for solving time-sensitive questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12775–12790.

Appendices

A Data Summary

Table 4 summarizes the dataset statistics used in this study. The numbers of official test samples are reported. Due to the budget, we evaluated Llama-3-8B on the full test set, GPT-4o-mini and Llama-3-70B on a random sample of up to 2,000 test set instances, and GPT-4o on 1,000 test set instances.

Additionally, the number of temporal relations considered in each dataset is included in Table 4. TempEvalQA-Bi and TRACIE focus mainly on the before-after relation. MCTACO includes diverse temporal relations, and the number of annotated candidates is reported. The questions in MCTACO are categorized into 5 question types, and examples for each type are provided in Figure 4.

	#Test	#Temporal relations
TempEvalQA-Bi	448	2
TRACIE	4248	2
MCTACO	9442	1-19

Table 4: Dataset Statistics. For TempEvalQA-Bi, the numbers represent the total number of questions. For TRACIE, the numbers refer to the number of story-hypothesis pairs. For MCTACO, the numbers reflect question-and-answer candidate pairs.

Event Duration
P. However, more recently, it has been suggested that it may date from earlier than Abdalonymus' death. Q. How long has it existed? (A) 2,000 hours (B) 2,000 years (C) 1 year (D) thousands of years (E) centuries (F) months
Event Frequency
P. Most of us have seen steam rising off a wet road after a summer rainstorm. Q. How often does it rain in the summer? (A) 333.33 times (B) every other minute (C) a couple times (D) every month (E) once a hour (F) once a year
Stationarity
P. She renews in Ranchipur an acquaintance with a former lover , Tom Ransome , now a dissolute alcoholic. Q. Is she still in Ranchipur? (A) yes (B) no
Event Ordering
P. Some of the people who took advantage of her through a questionable loan program were sent to jail. Q. What happened after they were put in jail? (A) they went to the store (B) they repented (C) even some people took these steps
Typical Time
P. Durer's father died in 1502, and his mother died in 1513. Q. When did Durer die? (A) 40 years later (B) 360 years later (C) 4545 (D) 40 seconds later (E) April 6, 1528

Figure 4: Examples of MCTACO Question Types. MCTACO covers various temporal aspects including event duration, frequency, stationarity, ordering, and typical time.

B Details of Evaluation Settings

This section outlines the detailed evaluation settings, including hyperparameters, resources, efficiency, and parsing methods. We use greedy decoding for **SP**, **CoT**, and **CCP**. For **Consistency**, **Reflect**, and **Debate**, we adopt the approach from Wang et al. (2023), employing top-k sampling with $k = 40$ and a temperature of 0.5 for the LLaMA model. For GPT-based models, we set the temperature to 0.7. **Consistency** samples 40 outputs from the decoder. **Reflect** refines the output iteratively for two iterations, including the initial output. In **Debate**, three agents engage in a debate over two rounds(Du et al., 2024). The implementations of the latter two baselines (**Reflect**, **Debate**) are based on the GitHub repository [1] from Du et al. (2024). Single-run performances are reported.

We note that our method prompts 3 times: for counterfactual question generation, counterfactual answer generation, and original question's answer generation, whose efficiency is compatible with or even more efficient than the three baselines. We also note that the Consistency baseline of Llama-3-70B cannot be reported due to its computation inefficiency.

For resources, we used the Transformers library (Wolf et al., 2020) and vLLM (Kwon et al., 2023) with 4 RTX A6000 GPUs for Llama-3 models. We used Openai API [2] for GPT models. For output parsing, the models generate the final answer after the phrase "`Final answer:`". Counterfactual exemplars are generated by modifying each dataset's questions, hypotheses, and candidate answers.

[1] https://github.com/composable-models/llm_multiagent_debate

[2] platform.openai.com

Models	Methods	MCTACO	
		EM	F1
GPT-4o -mini	MCQA-CoT	51.7	67.7
	CCP	**58.9**	**78.6**
GPT-4o	MCQA-CoT	65.6	73.3
	CCP	**66.2**	**80.2**

Table 5: Performance comparison on MCTACO with multiple-choice question answering setting.

C Further Analysis

C.1 Task generalizability

To demonstrate that our solution extends beyond binary question answering to multiple-choice question answering (MCQA), we evaluated the performance of GPT models using the original MCTACO evaluation setting (Zhou et al., 2019). While our primary evaluation decomposed the multiple-choice format into binary questions to measure inconsistency, it can be reconstructed for multiple-choice evaluation. We additionally introduced a baseline for MCQA (MCQA-CoT) that provides the context, question, and all candidate answers, generating one or more correct answers step-by-step. The results in Table 5 indicate that our method (CCP) outperforms the MCQA-CoT baseline on multiple-choice tasks, demonstrating its effectiveness in the MCQA setting.

C.2 Generated vs Retrieved Questions

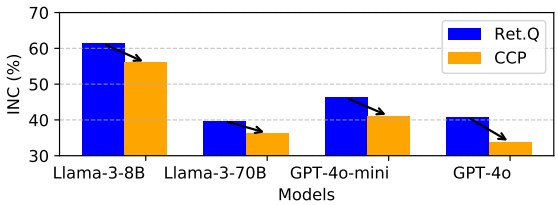


Figure 5: Comparison between counterfactual example collection methods on MCTACO with different models.

We tested whether our claim in Figure 2 can be generalized to other models. Figure 5 consistently confirms that creating counterfactual questions by generation handles diverse temporal relations better than retrieving questions across different models.

C.3 Number of Counterfactual Questions

We analyze the impact of varying the number of counterfactual questions on performance by testing with 1, 3, 5, and 7 questions. As shown in Figure 6, performance is highest with a single counterfactual question, with degradation observed as the number increases. The result aligns with findings from prior studies, where incorporating excessive counterfactual or contrastive information in prompts often results in diminished performance (Yao, 2024; Fang et al., 2024; Storaï and Hwang, 2024). For instance, multiple information degrade the performance in arithmetic and symbolic reasoning (Yao, 2024), multi-hop and commonsense question answering (Fang et al., 2024), and speculative decoding (Storaï and Hwang, 2024). These results indicate that such a phenomenon is not specific to our method but highlights a broader limitation in LLMs' ability to process multiple counterfactual information effectively.

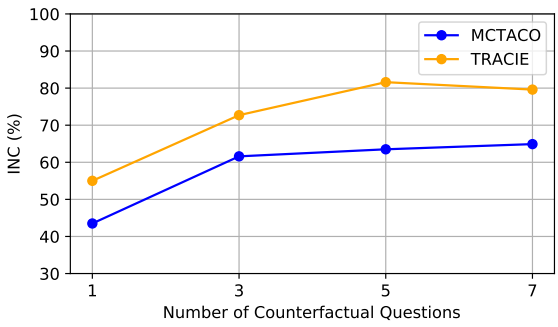


Figure 6: Inconsistency changes with the different number of counterfactual questions. The Llama-3-8B model is used.

		TimeQA		TimexNLI	
		ACC	F1	ACC	F1
Llama 3-8B	3 shot	**34.3**	40.8	68.0	65.3
	CoT 3 shot	32.3	38.4	**74.0**	**73.3**
	CCP 3 shot	34	**41.5**	67.3	62.2
GPT-4o -mini	3 shot	40	52.36	86.4	85.3
	CoT 3 shot	**43.3**	**56.75**	**90.4**	**90.3**
	CCP 3 shot	41	53.59	90.3	90.0

Table 6: Performance comparison on TimeQA and TimexNLI.

D Details of Limitations

D.1 Tasks with temporal indicators

Table 6 shows the experimental results for the tasks requiring the understanding of temporal indicators. We evaluated our method on TimeQA (Chen et al., 2021), the event-time ordering task, and TimexNLI-T1 (Thukral et al., 2021), the time-time ordering task, where CCP showed limited perfor-

mance gains.

D.2 Prompt Sensitivity

Our key decision in the prompt design was to separate the counterfactual question generation prompt (CCP), described in Appendix F.1, from the prompt that answers counterfactual and original questions, detailed in Appendix F.2. If we generate counterfactual questions and answers in an end-to-end manner using only the prompt in Appendix F.2 (CCP(e2e)), while it still outperforms CoT, the performance slightly decreases, as shown in Table 7.

E Usage of AI Assistants

ChatGPT was employed to generate answers in the prompt examples.

F Prompt Templates

We list the prompts that we used.

F.1 Prompt Templates for Generating Counterfactual Questions

To generate the counterfactual questions, we use the prompts provided below. We control question aspects by tailoring ICL examples and prompting the model to follow few-shot examples with the prompt: ("`following previous examples`").

TempEvalQA-Bi

[System]
You are an expert in temporal knowledge. Generate one related question to help determine the correctness of the original question, following the previous examples.
[User]
Question: Is Farkas sent into space on board the Soyuz before McBride on board the Orbiter Challenger?
[Assistant]
Is Farkas sent into space on board the Soyuz after McBride on board the Orbiter Challenger?
[User]
Question: Is McBride on board the Orbiter Challenger after Farkas was made a brigadier general?
[Assistant]
Is McBride on board the Orbiter Challenger before Farkas was made a brigadier general?
[User]
Question: Is Farkas was appointed military attache at the Hungarian embassy before he was made a brigadier?
[Assistant]
Is Farkas was appointed military attache at the Hungarian embassy after he was made a brigadier?

TRACIE

[System]
You are an expert in temporal knowledge. Generate one related hypothesis to help determine the plausibility of the original hypothesis, following previous examples.
[User]

Models	Prompts	TempEvalQA-Bi			TRACIE			MCTACO		
		ACC	F1	INC	ACC	F1	INC	ACC	F1	INC
Llama-3-8B	CoT	69.6	70.6	50.0	63.0	64.9	56.0	77.6	69.8	63.4
	CCP (e2e)	75.4	73.6	33.9	66.8	65.4	46.7	**83.1**	**81.0**	**53.7**
	CCP	**75.9**	**75.2**	**32.7**	**68.8**	**70.4**	**39.8**	78.7	81.7	57.7
GPT-4o-mini	CoT	81.3	79.9	29.0	73.2	68.5	42.7	80.9	73.7	58.9
	CCP (e2e)	80.8	80.2	27.7	79.0	78.3	28.0	87.1	84.4	52.1
	CCP	**88.8**	**88.7**	**19.6**	**82.5**	**81.2**	**20.2**	**87.8**	**85.8**	**42.1**

Table 7: Performance comparison with different prompts. The best and second-best performances are in boldface and underlined, respectively.

Hypothesis: Chad looked for his baseball cap starts after he got off the ride.
[Assistant]
Chad looked for his baseball cap starts before he got off the ride.
[User]
Hypothesis: Tom's friend says he is not gay starts before One day he noticed someone else asking
[Assistant]
Tom's friend says he is not gay starts after One day he noticed someone else asking
[User]
Hypothesis: Liza read a book on economics in the library starts before she got to the library
[Assistant]
Liza read a book on economics in the library starts after she got to the library

MCTACO

[System]
You are an expert in temporal commonsense knowledge. Generate related candidate answers that are either plausible or implausible that help determine the correctness of the original candidate answer following the previous examples.
[User]
Passage: The majority religion during the centuries of Ottoman rule, though a significant Christian minority remained. Question: What happened before Islam was the majority religion? Candidate answer: christianity was the majority religion.
[Assistant]
The Ottoman Empire had just begun / The spread of Byzantine influence in the region / The emergence of Islam as a minor religion
[User]
Passage: It's hail crackled across the comm, and Tara spun to retake her seat at the helm. Question: How long was the storm? Candidate answer: 6 years.
[Assistant]
an hour / a week / a month
[User]
Passage: Passage: About 30% of Ratners's profit already is derived from the U.S. Question: Is Ratners's profit derived from the U.S. today? Candidate answer: yes.
[Assistant]
no

F.2 Prompt Templates for CCP and Baselines

We only show the SP, CoT, and CCP since Consistency is the method that leverages CoT multiple times, and the Reflect and Debate follow the implementation of Du et al. (2024) after CoT.

Given the system and user message, the model generates an assistant's message. In our approach, the model is prompted twice: first to create an assistant's answer to the counterfactual question, and then to generate an assistant's answer to the original question. Though we use 3-shot examples, we provide 1 example for simplicity.

Prompt templates used to generate answers for the TempevalQA-Bi dataset are provided below.

TempevalQA-Bi: SP

[System]
You are an expert in temporal knowledge. Answer the question according to the article. Answer the question yes or no (Format: final answer: yes/no)
[User]
Article: Farkas, an air force captain, was sent into space on board the Soyuz 36 on May 26, 1980. He spent six days aboard the Salyut 6 spacecraft with three Soviet astronauts, Valery Kubasov, Leonid Popov and Valery Riumin. McBride, 54, of Lewisburg, West Virginia, was part of a seven-member crew aboard the Orbiter Challenger [...]
Question: Is Farkas sent into space on board the Soyuz before McBride on board the Orbiter Challenger?
[Assistant]
Final answer: yes

TempevalQA-Bi: CoT

[System]
You are an expert in temporal knowledge. Answer the question according to the article. Let's think step by step to answer the question. The answer should be "yes" or "no" followed by "Final answer: ".
[User]
Article: [...]
Question: Is Farkas sent into space on board the Soyuz before McBride on board the Orbiter Challenger?
[Assistant]
Let's think step by step. Farkas was sent into space on May 26, 1980 and McBride was on board the Orbiter Challenger in October 1984. May 26, 1980 is before October 1984. So the answer is: yes.
Final answer: yes

TempevalQA-Bi: CCP

[System]
You are an expert in temporal knowledge. Answer the question according to the article. To improve understanding, follow these steps: Answer the related question to better understand the timeline (Format: answer: [yes/no]) Using the answers to the related questions as the additional information, answer the original question yes or no (Format: final answer: [yes/no]).
[User]
Article: [...]
Original question: Is Farkas sent into space on board the Soyuz before McBride on board the Orbiter Challenger?
Related question: Is Farkas sent into space on board the Soyuz after McBride on board the Orbiter Challenger?
Answer the related question
[Assistant] *# Prediction for the counterfactual question*
Farkas was sent into space on board the Soyuz 36 on May 26, 1980. McBride was part of the crew on board the Orbiter Challenger in October 1984. Since 1980 is before 1984, no, Farkas was not sent into space after McBride.
answer: no
[User]
Now, answer the original question. Question: Is Farkas sent into space on board the Soyuz before McBride on board the Orbiter Challenger?
[Assistant] *# Prediction for the original question*
Farkas was sent into space on May 26, 1980 and McBride was on board the Orbiter Challenger in October 1984. May 26, 1980 is before October 1984. So the answer is: yes.
Final answer: yes

Prompt templates used to generate answers for the TRACIE dataset are provided below.

TRACIE: SP

[System]
You are an expert in temporal knowledge. You can understand the implicit temporal relationships between events to make judgments. Your task is to determine the relationship between the story and hypothesis. A 'positive' label means the hypothesis can be inferred from the story or logically sound. A 'negative' label means the hypothesis cannot be inferred from the story. You must answer only with the labels 'positive' or 'negative'. (Format: final answer: posi-

tive/negative)
[User]
Story: Chad looked for his baseball cap starts after he got off the ride story: Chad had gone to an amusement park. He was riding on the roller coaster.
Hypothesis: Chad looked for his baseball cap starts after he got off the ride.
[Assistant]
Final answer: positive

TRACIE: CoT

[System]
You are an expert in temporal knowledge. You can understand the implicit temporal relationships between events to make judgments. Your task is to determine the relationship between the story and hypothesis. A 'positive' label means the hypothesis can be inferred from the story or logically sound. A 'negative' label means the hypothesis cannot be inferred from the story. You must give final answer only with the labels 'positive' or 'negative' after the "Final answer:" prompt. Let's think step by step to evaluate the relationship.
[User]
Story: [...]
Hypothesis: Chad looked for his baseball cap starts after he got off the ride.
[Assistant]
Let's think step by step:
The story states that Chad's baseball cap fell off during the roller coaster ride.
It also mentions that Chad found the cap after he got off the ride. To find something, one typically needs to look for it first. Since Chad found the cap after getting off the ride, he must have started looking for it after getting off the ride. The hypothesis directly aligns with this sequence of events.
Final answer: positive

TRACIE: CCP

[System]
You are an expert in temporal knowledge. You can understand the implicit temporal relationships between events to make judgments. Your task is to determine the relationship between the story and hypothesis. A 'positive' label means the hypothesis can be inferred from the story or logically sound. A 'negative' label means the hypothesis cannot be inferred from the story. You must give final answer only with the labels 'positive' or 'negative'. To improve understanding, follow these steps: Evaluate the related hypothesis to help clarify the story's timeline (Format: answer: [positive/negative]). Using answers to the related hypothesis as the additional information, evaluate the original hypothesis (Format: final answer: [positive/negative]).
[User]
Story: [...]
Hypothesis: Chad looked for his baseball cap starts after he got off the ride.
Related hypothesis: Chad looked for his baseball cap starts before he got off the ride.
Answer the related hypothesis
[Assistant] *# Prediction for the counterfactual question*
The hypothesis conflicts with the story, as Chad finds the cap after the ride.
Answer: negative
[User]
Now, evaluate the original hypothesis: Chad looked for his baseball cap starts after he got off the ride.
[Assistant] *# Prediction for the original question*
Chad rides the roller coaster while wearing a baseball cap, which falls off during the ride. After getting off the ride, Chad finds the cap. The hypothesis suggests that Chad started looking for the cap after getting off the ride, which aligns with the story's sequence of events.
Final answer: positive

Prompt templates used to generate answers for the MCTACO dataset are provided below.

MCTACO: SP

[System]
You are expert in temporal commonsense knowledge. Your goal is to determine if the candidate answer to a given question is plausible based on the information in

the passage. Answer the related candidate answers then answer the candidate answer. Provide a 'yes' or 'no' response to indicate whether the original candidate answer is correct. (Format: final answer: yes/no) If the passage does not provide explicit details, provide an educated guess based on what is common or plausible in similar situations

[User]

Passage: the majority religion during the centuries of Ottoman rule, though a significant Christian minority remained.

Question: What happened before Islam was the majority religion?

Candidate answer: the end of white-minority rule.

[Assistant]

Final answer: no

MCTACO: CoT

[System]

You are an expert in temporal commonsense knowledge. Your goal is to determine if the candidate answer to a given question is plausible based on the information in the passage. Let's think step by step and provide a final 'yes' or 'no' response to indicate whether the original candidate answer is correct. If the passage does not provide explicit details, provide an educated guess based on what is common or plausible in similar situations. The final answer should be in the format: "Final answer: yes" or "Final answer: no".

[User]

Passage: the majority religion during the centuries of Ottoman rule, though a significant Christian minority remained.

Question: What happened before Islam was the majority religion?

Candidate answer: christianity was the majority religion

[Assistant]

Let's think step by step.

Step 1: Analyze the passage. The passage states that during the centuries of Ottoman rule, Islam was the majority religion, while a significant Christian minority remained. This implies that before the Ottoman Empire (which was Islamic), there could have been a different majority religion.

Step 2: Historical context. Before the rise of the Ottoman Empire, large parts of the Middle East, North Africa, and parts of Europe were under the rule of the Byzantine Empire, which was a Christian empire. Additionally, many regions that later came under Ottoman rule were predominantly Christian.

Step 3: Plausibility of the candidate answer. Given that Christianity was the dominant religion in many regions before the Ottoman Empire and its Islamic rule, it is plausible that Christianity was the majority religion before Islam in those areas.

Final answer: yes

MCTACO: CCP

[System]

You are an expert in temporal commonsense knowledge. Your goal is to determine if the candidate answer to a given question is plausible based on the information in the passage. To improve understanding, follow these steps: generate related candidate answers that are either plausible or implausible (Format: answer: yes/no) to help in determining the correctness of the original candidate answer. Provide a 'yes' or 'no' response to indicate whether the original candidate answer is correct (Format: final answer: yes/no). If the passage does not provide explicit details, provide an educated guess based on what is common or plausible in similar situations

[User]

Passage: the majority religion during the centuries of Ottoman rule, though a significant Christian minority remained.

Question: What happened before Islam was the majority religion?

original candidate answer: christianity was the majority religion

Related candidate: the end of white-minority rule

Can the candidate answer the given question? yes or no

[Assistant] *# Prediction for the counterfactual question*

This is incorrect because the passage dis-

cusses events during Ottoman rule, not before it.
answer: no
[User]
Now, can the candidate answer the given question? candidate answer: christianity was the majority religion
[Assistant] *# Prediction for the original question*
This is plausible because Islam would have started to emerge before becoming the majority.
Final answer: yes

G Scientific Artifacts

We used existing scientific artifacts for research purposes, and the use of existing artifacts was consistent with their intended applications.

TempEvalQA-Bi: MIT license
TRACIE: Apache-2.0 license
Llama-3: custom commercial license
OpenAI API: Apache-2.0 license

Author Index