

KnowFM 2026

**The 4th Workshop on Towards Knowledgeable Foundation  
Models**

**Proceedings of the Workshop**

July 3, 2026

©2026 Association for Computational Linguistics

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](mailto:acl@aclweb.org)

ISBN 979-8-89176-403-3

## Preface

Welcome to the 4th Workshop on Towards Knowledgeable Foundation Models (KnowFM 2026), co-located with ACL 2026 and held in San Diego, California, United States, on July 3, 2026.

Foundation models have become vast repositories of world and task knowledge, yet how that knowledge is acquired, stored, retrieved, and faithfully used remains only partially understood. KnowFM is organized around the lifecycle of knowledge in foundation models, spanning four stages: how knowledge *emerges* during pre-training, how it is *injected* from external sources, how it is *updated* or edited over time, and how it is *probed and generated* at inference. The workshop brings together research on the analysis of knowledge in foundation models (how much they know, where that knowledge comes from, and how it is represented), knowledge injection during training and inference, retrieval-augmented generation, knowledge updating and editing, knowledge extraction and distillation, hallucination mitigation and the evaluation of faithfulness, knowledge conflicts between parametric memory and retrieved or long-context information, and multimodal and agentic knowledge systems.

This fourth edition of the workshop accepted 38 papers. The present volume collects the 14 accepted papers whose authors elected for archival inclusion in the ACL Anthology proceedings; the remaining accepted papers were presented at the workshop on a non-archival basis. We are also delighted to host invited talks by Sewon Min (University of California, Berkeley), Eunsol Choi (New York University), Mohit Iyyer (University of Maryland), Danqi Chen (Princeton University), Yulia Tsvetkov (University of Washington), and Luna Dong (Meta).

We are grateful to the 72 members of our Program Committee, whose careful and timely reviews made the program possible, and to our Advising Committee, Heng Ji, Mohit Bansal, Isabelle Augenstein, and Jiawei Han, for their guidance. We thank the authors for their contributions, the invited speakers for sharing their perspectives, and the ACL 2026 organizers and publication chairs for their support. We hope you find the workshop stimulating.

The KnowFM 2026 Organizing Committee

# Organizing Committee

## Organizing Committee

Canyu Chen, Northwestern University  
Yuji Zhang, University of Illinois Urbana-Champaign  
Zoey Sha Li, Amazon  
Zihan Wang, Northwestern University  
Qineng Wang, Northwestern University  
Jinyan Su, Cornell University  
Priyanka Kargupta, University of Illinois Urbana-Champaign  
Sara Vera Marjanović, University of Copenhagen  
Jeff Z. Pan, University of Edinburgh  
Manling Li, Northwestern University

## Publication Chair

Canyu Chen, Northwestern University

## Advising Committee

Heng Ji, University of Illinois Urbana-Champaign  
Mohit Bansal, University of North Carolina at Chapel Hill  
Isabelle Augenstein, University of Copenhagen  
Jiawei Han, University of Illinois Urbana-Champaign

## Program Committee

### Program Committee

Shivam Adarsh, Copenhagen University  
Atul Anand, curiouskalp  
Ashwin Aravind, Indian Institute of Technology, Bombay  
Nikolay Banar, Universiteit Antwerpen  
Prakhar Bansal, OpenAI  
Mina Basirat, University of Central Florida  
Aprameya Bharadwaj, Adobe Systems  
Zacharie Bugaud, Astera Institute  
Yisu Chen, Wilfrid Laurier University  
Gaurav Chodwadia, Walmart  
Lu Dai, Hong Kong University of Science and Technology  
Xiaoyu Deng, University of Pennsylvania  
Pranamy Nilesh Deshpande, Gokhale Education Society's R. H. Sapat College of Engineering,  
Management Studies and Research  
Sahil Rajesh Dhayalkar, Brain Corporation  
Liang Ding, Alibaba Group  
Yixuan Du, Georgetown University  
Ritvik Garimella, University of South Carolina  
Dipanwita Guhathakurta, International Business Machines  
Akshat Gupta, University of California, Berkeley  
Joanna Hao, University of Alberta  
Mohd Ariful Haque, Clark Atlanta University  
Tianyi Huang, Cleanlab  
Shabbir Hussain, Amazon  
Parin Rajesh Jhaveri, J.P. Morgan Chase  
Adam Kamel, University of Waterloo  
Gael Kengmegni, Independent  
Yoav Kor, Tel Aviv University  
Ashish Balkishan Lathkar, Hemut Co  
Qi Li, Tsinghua University  
Zichao Li, University of Waterloo  
Duoduo Liao, George Mason University  
Shiyang Liu, Huazhong University of Science and Technology  
Devarshi Mahajan, Northeastern University  
Zhaohan Meng, University of Glasgow  
Noor Islam S. Mohammad, Istanbul Technical University  
Yi Nian, University of Southern California  
Michael-Andrei Panaitescu-Liess, University of Maryland, College Park  
Rahul Pandya, DePaul University  
Hari Charan Panjwani, Facebook  
Maitrik Patel, Apple  
Pankayaraj Pathmanathan, University of Maryland, College Park  
Viswanathan Ranganathan, IEEE  
Ravi Ranjan, Florida International University  
Shivam Ratnakar, University of Southern California  
Pravish Sainath, Montreal Institute for Learning Algorithms, University of Montreal, University of

Montreal  
Kanchan Sarkar, Tiktok  
Krishna Senthilkumar, Algoverse AI Research  
Eugenie Shi, Stanford University  
Jiazhao Shi, Amazon  
Shuzheng Si, Tsinghua University  
Malaiappan Srinivasan, University of Zurich  
Elena Stringli, National Technical University of Athens  
Hexiang Tan, Chinese Academy of Sciences  
Wout Teillers, University of Groningen  
Rohith Uppala, LinkedIn  
Matias Valdenegro-Toro, University of Groningen  
Tomás Vergara Browne, McGill University  
Yike Wang, Department of Computer Science, University of Washington  
Zhaohui Geoffrey Wang, University of Southern California  
Juesi Xiao, Tianjin University  
Zhichao Xu, Amazon  
Shuo Yang, University of Melbourne  
Yuqing Yang, University of Southern California  
Haeun Yu, University of Copenhagen  
Wenhao Yuan, The University of Hong Kong  
Wajdi Zaghoulani, Northwestern University  
Pardis Sadat Zahraei, University of Illinois at Urbana-Champaign  
Fengwen ZHAI, Lanzhou Jiaotong University  
Wenjing Zhai, The High School Affiliated to Beijing Normal University  
Weizhi Zhang, University of Illinois at Chicago  
Jianing Zhao, Tsinghua University  
Yuanjie Zhu, University of Illinois at Chicago

## Table of Contents

<i>Annotation Frameworks Shape Model Knowledge: Safety Alignment in Large Language Models</i> Wajdi Zaghouani .....	1
<i>Blind Single-Layer Activation Edits Show a Break/Fix Asymmetry in Factual Recall</i> Zacharie Bugaud .....	13
<i>What Does Alignment Cost? The Structural Brittleness of Chain-of-Thought Reasoning</i> Joanna Hao, Shanduojiang Jiang and Sai Asish Nakka .....	25
<i>bLLeQA: Benchmarking LLMs for Grounded Legal Question-Answering in French and Dutch</i> Nikolay Banar, Ehsan Lotfi, Jens Van Nooten, Marija Kliocaite and Walter Daelemans .....	34
<i>VLA-Forget: Vision-Language-Action Unlearning for Embodied Foundation Models</i> Ravi Ranjan and Agoritsa Polyzou .....	60
<i>Overcoming the Impedance Mismatch: A Theoretical Roadmap for Fusing Foundation Models and Knowledge Graphs</i> Sahil Rajesh Dhayalkar .....	78
<i>LLM-MemCluster: Empowering Large Language Models with Dynamic Memory for Text Clustering</i> Yuanjie Zhu, Liangwei Yang, Ke Xu, Weizhi Zhang, Zihong Song, Jindong Wang and Philip S. Yu	90
<i>Test-Time Training for Zero-Resource Dense Retrieval Reranking</i> Shiyuan Liu and Yichen Li .....	105
<i>Multimodal Generative Engine Optimization: Rank Manipulation for Vision–Language Model Rankers</i> Yixuan Du, Chenxiao Yu, Haoyan Xu, Ziyi Wang, Yue Zhao and Xiyang Hu .....	115
<i>Beyond Retrieval: Bi-Temporal State Arbitration for Longitudinal Healthcare Agents</i> Jianing Zhao, Xiaoquan Zhi and Xinqiang Yu .....	129
<i>RSCE: Training-Free Residual Stream Encoding for Persistent Context Amortization</i> Adam Kamel and Eric Xu .....	138
<i>Tricking Open-World Object Recognition Models: Uncertainty in Out-of-Distribution Detection</i> Wout Teillers and Matias Valdenegro-Toro .....	147
<i>Knowledge Localization and Editability in Small Language Models: A Multi-Stage Experimental Study</i> Pranamy Niles Deshpande, Aiswarya Konavoor and Sreedath Panat .....	165
<i>One Retrieval to Cover Them All: Co-occurrence-Aware Knowledge Base Reorganization for Session-Level RAG</i> Shivam Ratnakar, Yixuan Zhu, Cecilia Cheng and Chaya Vijayakumar .....	173

# Annotation Frameworks Shape Model Knowledge: Safety Alignment in Large Language Models

Wajdi Zaghouani

Communication Program

Northwestern University in Qatar

Doha, Qatar

wajdi.zaghouani@northwestern.edu

## Abstract

Large language models (LLMs) are commonly described as acquiring knowledge through large scale pretraining on textual corpora. This view underestimates the epistemic consequences of post training safety mechanisms. Modern LLMs undergo extensive safety alignment via curated datasets, human annotations, and reinforcement learning from human feedback (RLHF), processes that do not merely constrain outputs but actively reshape how propositional and procedural knowledge is accessed and expressed. We propose a conceptual framework in which safety alignment functions as a systematic form of knowledge editing at scale. Annotation frameworks used to construct safety datasets act as normative ontologies that partition language into categories of acceptable and unacceptable content, and alignment training propagates these distinctions into model behaviour. We introduce the Safety Knowledge Pipeline (SKP), a four stage framework describing how pretraining knowledge is progressively filtered, reframed, and constrained through annotation and alignment mechanisms. We identify three mechanisms of knowledge modification, suppression, reframing, and substitution, each with distinct diagnostic signals, and we operationalise them in a cross lingual evaluation protocol. Throughout, we distinguish carefully between behavioural claims that follow from prior empirical literature and representational claims that remain open hypotheses. Case studies spanning harmful instruction queries, hate speech annotation in Arabic dialects, and culturally variable discourse illustrate the framework. We further discuss how treating annotator disagreement as a training signal rather than noise can mitigate the culturally hegemonic effects of current alignment pipelines.

## 1 Introduction

Large language models have become central infrastructure for natural language processing, exhibiting remarkable capacity to store and express factual,

commonsense, and procedural knowledge (Petroni et al., 2019; Brown et al., 2020). This has motivated a research programme examining where that knowledge comes from, how reliably it is expressed, and whether it can be selectively modified (Meng et al., 2022; De Cao et al., 2021; Ji et al., 2023).

A prevalent assumption in this programme treats knowledge acquisition as essentially complete at the pretraining stage. Post training processes such as instruction tuning and safety alignment are then understood primarily as behavioural constraints layered on top of already formed knowledge representations. The model knows what it knows; safety training governs only what it says.

This paper argues that the assumption is incomplete. Safety alignment, as currently practiced, does not merely constrain the expression of knowledge but also restructures how knowledge is organised, accessed, and expressed. We use the term *knowledge* in a deliberately scoped sense focused on the classes of content that alignment pipelines explicitly target. This is largely procedural content (instructions, methods, advice) and culturally or normatively loaded propositional content, rather than non pluralistic facts such as standard encyclopaedic entries. The framework still applies to non pluralistic facts when annotation policies happen to touch them, but we do not claim that all factual associations are reshaped uniformly by alignment, and a reviewer correctly observed that the strongest effects are on content that interacts with the normative categories defined by annotation. The restructuring we describe is mediated by annotation frameworks, the guidelines that human annotators use when constructing safety training datasets. These guidelines do not passively reflect pre existing facts about harm; they construct normative ontologies that partition language into policy determined categories. When models are trained to optimise reward signals derived from these annotations, they internalise the ontological distinctions

encoded in the guidelines in ways that alter how parametric knowledge is exposed at inference time.

We are careful throughout to distinguish what is shown from what is hypothesised. The behavioural effects discussed in this paper (the suppression, reframing, and substitution patterns observed at inference) are supported by published empirical work. The stronger claim that alignment modifies parametric storage in the same mechanistic sense as targeted editing methods such as ROME (Meng et al., 2022) remains an open hypothesis. We mark this distinction at each step rather than collapsing the two.

This argument connects several threads that have not previously been examined together: the mechanics of parametric knowledge editing (Meng et al., 2022, 2023), inference time representation control through steering vectors (Zou et al., 2023), the epistemic effects of annotation design choices (Röttger et al., 2022; Davani et al., 2022), the cultural specificity of aligned model behaviour (Sanurkar et al., 2023), and empirical evidence that alignment induces systematic bias against dialectal varieties (Robinson et al., 2025). Placing these findings within a unified framework shows they are not isolated anomalies but systematic consequences of treating alignment as a process of knowledge construction.

Concretely, this paper makes four contributions. We introduce the *Safety Knowledge Pipeline* (SKP), a four stage framework describing how pretraining knowledge is progressively shaped by annotation and alignment. We identify and characterise three mechanisms of alignment induced knowledge modification (suppression, reframing, and substitution) with tighter operationalisations than prior work. We provide a diagnostic taxonomy with automatic proxy metrics for each mechanism, illustrated by a worked example. And we propose a cross lingual evaluation protocol designed to surface annotation framework induced knowledge boundary mismatches across languages and cultures, including a discussion of how pluralistic alignment can mitigate these effects.

We position the paper explicitly as a conceptual contribution with a concrete evaluation agenda. The empirical case for the framework rests on synthesising prior published results, especially the AL-QASIDA evaluation of nine LLMs across eight Arabic dialect varieties (Robinson et al., 2025), which provides direct behavioural evidence that post training induces systematic dispreference for

under resourced language varieties. Implementing the bypass probing and framing analysis components of the protocol is the natural next step and is left to future work.

## 2 Background and Related Work

**Knowledge in LLMs.** We focus the discussion on two classes of content that alignment pipelines explicitly target: procedural knowledge (instructions, methods, advice) and propositional content that interacts with normative categories (claims about harm, identity, culture, health, politics, and similar domains). Pretraining primarily shapes both classes through statistical association; safety alignment then introduces an additional normative layer that interacts with and reshapes downstream expression. As Reviewer XLfk correctly noted in review, we do not claim that all propositional knowledge is reshaped uniformly. Non pluralistic facts (e.g., capital cities, mathematical identities) are typically not the target of safety annotation, and where they are touched it is usually incidental to other categories (e.g., facts entangled with culturally contested narratives).

**Parametric knowledge and its distribution.** Petroni et al. (2019) demonstrated that LMs can answer factual cloze style queries without retrieval. Roberts et al. (2020) showed that scaling substantially increases factual recall. Kandpal et al. (2023) demonstrated that parametric knowledge is unevenly distributed: facts appearing rarely in training corpora are stored unreliably, creating a long tail gap that interacts with alignment.

**Knowledge editing.** De Cao et al. (2021) proposed constrained fine tuning for factual edits. Meng et al. (2022) introduced ROME, which localises factual associations to specific feed forward layers and overwrites them with rank one updates; Meng et al. (2023) extended this to MEMIT for batch editing. Mitchell et al. (2022) proposed SERAC, a retrieval augmented editing approach. These methods target precise, semantically specific edits to discrete factual associations, often non pluralistic ones (e.g., changing the answer to “Who is the prime minister of the UK?”). We invoke ROME and MEMIT only as a conceptual reference point for the idea that model knowledge can be intentionally modified after pretraining, not as a claim that alignment operates through the same mechanism. Reviewer XLfk correctly observed that this

distinction matters: alignment is unlikely to modify non pluralistic facts the way ROME does, and the kind of content reshaped by alignment is mostly normative and pluralistic.

**Inference time representation control.** A more directly relevant body of work is the literature on steering vectors and representation engineering. [Zou et al. \(2023\)](#) showed that high level concepts such as harmfulness, honesty, and power seeking can be extracted as directions in activation space and used to read or steer model behaviour at inference time without modifying weights. This work is methodologically closer to the phenomena we describe: it operates at the level of representations that govern when and how the model expresses normative content, rather than on discrete factual associations. We treat steering vector findings as evidence that the kind of content alignment shapes (normative, attitudinal, instructional) is plausibly encoded as distributed representational structure, and as a candidate toolkit for empirically probing the SKP.

**Hallucination and factuality.** [Ji et al. \(2023\)](#) survey a large literature on LLMs generating plausible but unsupported content. [Maynez et al. \(2020\)](#) showed abstractive summarisation models routinely introduce unsupported propositions. [Lin et al. \(2022\)](#) introduced TruthfulQA for evaluating factual accuracy on misconception eliciting questions. Our framework complements this literature: safety alignment introduces a distinct source of factual distortion, not confabulation from distributional pressure but selective suppression and reframing guided by normative annotation.

**RLHF and safety alignment.** [Ouyang et al. \(2022\)](#) showed that RLHF substantially improves instruction following and reduces harmful outputs. [Bai et al. \(2022a\)](#) analysed trade offs between helpfulness and harmlessness. [Bai et al. \(2022b\)](#) introduced Constitutional AI (CAI), reducing dependence on direct human annotation. [Rafailov et al. \(2023\)](#) proposed DPO, simplifying alignment training. Across all these approaches, normative annotation categories (whether produced by humans or by AI following written principles) remain the epistemic foundation.

**Annotation design and pluralism.** [Röttger et al. \(2022\)](#) distinguished prescriptive from descriptive annotation, showing the two paradigms produce systematically different models. [Davani et al.](#)

[\(2022\)](#) demonstrated that annotator disagreement on toxicity reflects genuine social variation rather than noise. [Waseem et al. \(2018\)](#) argued that hate speech datasets import assumptions from legal domains that do not transfer to computational settings. This body of work establishes that annotation frameworks are not neutral transcriptions of social reality; they are, in the terminology we introduce below, normative ontologies.

**Arabic NLP and post training bias.** Arabic poses particular challenges for safety annotation because of its diglossia and dialectal diversity ([Habash, 2010](#)). Work on Arabic error annotation has documented the difficulty of developing guidelines that transfer across regional varieties ([Zaghrouani et al., 2014](#)). Arabic hate speech datasets include L-HSAB for Levantine Arabic ([Mulki et al., 2019](#)) and a dataset targeting religious hate speech in Arabic Twitter ([Albadi et al., 2018](#)). Most directly, [Robinson et al. \(2025\)](#) evaluate nine LLMs across eight Arabic dialect varieties and find that post training makes models measurably more reluctant to generate dialectal Arabic (DA), even when the models understand DA well. Few shot dialectal examples partially repair this bias. This is direct behavioural evidence that post training alignment introduces systematic dispreference for under resourced language varieties, a concrete instantiation of the knowledge boundary mismatch we theorise.

### 3 Annotation Frameworks as Normative Ontologies

Safety alignment depends on large scale human annotation in which annotators classify prompts and responses according to categories defined by annotation guidelines. We argue these guidelines function as normative ontologies: they partition the space of possible linguistic expressions into categories that the alignment system subsequently internalises.

Unlike factual ontologies (taxonomies of entity types used in information extraction), safety annotation guidelines carry explicit evaluative content. A category such as “violent speech” or “medical misinformation” does not merely describe a linguistic property; it encodes a judgement about harm, responsibility, and the appropriate response. This normative loading has three consequences relevant to our framework.

First, category boundaries are constructed rather than discovered. Researchers, ethicists, and legal

experts define where “offensive” ends and “illegal” begins, where “discussing violence” differs from “inciting violence”. As Röttger et al. (2022) document, these decisions vary across frameworks and have measurable downstream effects on model behaviour.

Second, category membership is culturally variable. Expressions that constitute insults or incitement in one community may carry different valence in another. Annotation frameworks that do not account for this variability encode culturally specific assumptions as universal constraints (Davani et al., 2022; Santurkar et al., 2023).

Third, the normative content of annotation categories is not recoverable from the model. A model trained to suppress a category of queries does not encode the normative reasoning behind suppression; it encodes a statistical association between surface patterns and a reward or penalty signal. The epistemic consequences of annotation design are therefore difficult to audit from the model itself, creating a transparency problem that governance frameworks must address.

## 4 The Safety Knowledge Pipeline

We propose the Safety Knowledge Pipeline (SKP) as a framework for understanding how pretraining knowledge is progressively shaped by annotation and alignment. Figure 1 illustrates the four stages. The framework’s added value, beyond restating existing critiques of RLHF and annotation bias, is twofold. First, it explicitly separates the loci at which knowledge is shaped (annotation, training, inference), which is necessary for attributing observed behavioural effects to specific design choices. Second, it pairs each stage with a class of intervention (annotation reform, training procedure changes, inference time moderation), which gives the framework prescriptive purchase that unstructured critiques lack.

**Stage 1.** Pretraining on large corpora yields content of the kinds described above. The form of this content as “parametric knowledge” follows Petroni et al. (2019).

**Stage 2.** Human annotators classify content according to guidelines defining harmful or undesirable speech. These frameworks encode normative judgements about what constitutes harm, who may be harmed, and in what context. They function as knowledge ontologies that categorise language

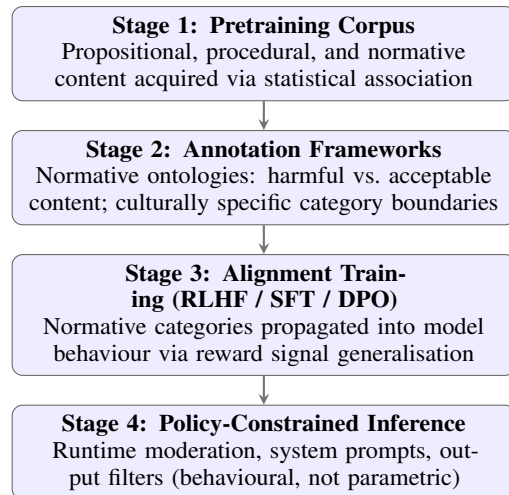


Figure 1: The Safety Knowledge Pipeline (SKP). Each stage introduces additional normative constraints. Stages 1 to 3 affect what knowledge the model can express through training; Stage 4 applies behavioural constraints at inference time. Distinguishing training time (Stage 3) from inference time (Stage 4) effects is key to understanding suppression persistence and bypass asymmetries, and is the natural axis along which the framework can be empirically tested.

according to social and ethical criteria (Section 3).

**Stage 3.** Through RLHF, SFT, CAI, or DPO (Ouyang et al., 2022; Bai et al., 2022b; Rafailov et al., 2023), the model learns to approximate the reward signal derived from annotator judgements. This stage propagates the categorical distinctions of the annotation framework into model behaviour. Whether this propagation reaches into the parameters in the same mechanistic sense as ROME style edits is, as Reviewer cien correctly observed, an open empirical question. The behavioural fingerprint is clear: aligned models systematically refuse, reframe, or substitute on prompt classes that map to annotation categories, and these patterns are recoverable by light fine tuning (Yang et al., 2023). Whether the underlying representational change is concentrated in particular layers or distributed across the residual stream is the kind of question that representation level methods, including steering vectors (Zou et al., 2023) and contrast based probes (Burns et al., 2022), are well suited to investigate. We treat this localisation question as an empirical agenda the framework licences rather than a claim it establishes.

**Stage 4.** At inference time, system prompts, moderation classifiers, and output filters apply additional constraints that are behavioural rather than

learned. Distinguishing Stage 3 (training time) from Stage 4 (inference time) effects is methodologically important because it identifies the intervention point for any observed failure. The cleanest experimental design is to compare open weight model pairs at varying stages of post training (base, SFT only, SFT + RLHF) with Stage 4 filters ablated. This isolates the contribution of alignment training itself from runtime moderation and is, in our view, the most tractable next step for grounding the framework empirically.

## 5 Three Mechanisms of Knowledge Modification

We identify three mechanisms through which safety alignment modifies knowledge behaviour. The mechanisms are defined first in behavioural terms (what the model does at inference time, which is directly observable) and then linked to underlying causes (what changed during training to produce that behaviour). This separation responds directly to Reviewer XLfk’s observation that earlier framings collapsed two distinct ideas. To make this concrete: Reviewer XLfk asked what differentiates “knowledge in model parameters conditionally inaccessible” from “does not engage with the substance of the query”. The answer is that the former is the internal state, while the latter is one of several possible observable outputs from that state. Suppression and substitution can both arise from inaccessible knowledge but produce different behavioural signatures: suppression produces a refusal that the model itself recognises as such (“I cannot help with that”), while substitution produces an affirmative response that does not address the asked content (a safety redirect, professional referral, or topic change). The diagnostic taxonomy in Section 6 operationalises this distinction.

### 5.1 Knowledge Suppression

**Definition.** Suppression occurs when, conditional on a prompt that maps to a suppressed category, the model declines to produce content it would otherwise be capable of producing. The behavioural signature is an explicit refusal (“I cannot help with that”, “I am unable to provide such content”).

**Evidence.** Perez et al. (2022) showed that LM generated red teaming prompts elicit harmful outputs from aligned models at rates substantially above zero, with bypass patterns consistent across model families. This demonstrates that suppres-

sion is implemented as statistical pattern matching over prompt distributions: equivalent content arriving through a different surface form often elicits the suppressed response. The empirical behaviour, refusal under one surface form and answer under another, does not require a strong claim about parametric storage. Yang et al. (2023) demonstrated that fine tuning on roughly 100 harmful examples with one GPU hour can subvert safety alignment in open weight models, providing further evidence that whatever parametric change alignment effects is, at least in open models, shallow and reversible.

**Knowledge asymmetry.** Because suppression is distributional rather than absolute, users with sufficient prompt engineering sophistication can access nominally suppressed information while others cannot. This is a knowledge access asymmetry with equity implications that the safety literature has only partially addressed.

**Diagnostic signal.** Suppression is detected by bypass probing: generating semantically equivalent paraphrases and cross lingual variants of a target query and measuring bypass rate. High bypass rate indicates that suppression is implemented shallowly (essentially as a prompt classifier) rather than as a robust parametric change.

### 5.2 Knowledge Reframing

**Definition.** Reframing occurs when the model produces a response that addresses the asked content but selectively emphasises or omits material in ways that reflect the normative commitments of the annotation framework. The behavioural signature is an affirmative, on topic response that nonetheless diverges systematically from a culturally appropriate reference.

**Evidence.** Santurkar et al. (2023) showed that aligned LLMs express opinions that cluster in culturally specific ways reflecting the demographic distribution of annotators, demonstrating systematic normative skew introduced by annotation frameworks and propagated through alignment. As a concrete illustration: when asked about traditional culinary practices, medicinal herbalism, or historical political movements that hold different valences in different communities, aligned models tend to produce responses calibrated to the safety norms of the annotation framework rather than to the user’s actual context. This produces factual incompleteness without triggering a refusal.

**Diagnostic signal.** Reframing is detected by attribute coverage divergence: comparing salient

attribute coverage in model responses against a reference knowledge base (e.g., structured encyclopaedic sources or scholarly literature) using automated content selection metrics and framing lexicons. Unlike suppression, which produces a null response, reframing requires a reference response for comparison. Inter annotator agreement on “complete vs. reframed” should be reported and treated as a key reliability check.

### 5.3 Knowledge Substitution

**Definition.** Substitution occurs when the model replaces a substantive response with policy compliant content that does not engage with the asked query. Unlike reframing, which modifies content while staying on topic, substitution swaps in a categorically different kind of output: a safety warning, a disclaimer, a referral to a professional, or a redirect to other resources. The behavioural signature is an affirmative response that is off topic with respect to the question asked.

**Evidence.** Substitution is particularly consequential in medical, legal, and cultural domains. [Bender et al. \(2021\)](#) note that the costs and benefits of safety mechanisms are not uniformly distributed: restrictions that are acceptable inconveniences for well resourced users may represent meaningful barriers for others.

**Concrete example.** A user asks the model how to manage a specific medication interaction. Three possible responses are: (a) a refusal (“I cannot discuss medication interactions”), which is suppression; (b) a partial answer covering some interactions while omitting the relevant one in a way that diverges from a clinical reference, which is reframing; (c) an affirmative answer that is off topic (“You should consult a healthcare professional about medication interactions. Here are some general principles of safe medication use . . .”), which is substitution. The three response types produce different behavioural signatures and require different diagnostics.

**Automatic proxy.** Substitution can be detected automatically as a first pass filter using a disclaimer or safety redirect classifier. High recall on explicit safety language (“consult a professional”, “I cannot provide”) identifies clear substitution cases, which can then be verified against reference responses to confirm that the response is genuinely off topic rather than merely cautious.

## 6 Taxonomy of Knowledge Modification Patterns

Table 1 synthesises the three mechanisms into a diagnostic taxonomy with operationalisable indicators.

The taxonomy surfaces several important properties. First, the mechanisms differ in their detectability: suppression produces a clear null signal, while reframing and substitution produce affirmative responses that require a reference to identify as incomplete or off topic. Second, they interact with model scale differently: suppression may become more consistent at larger scales as the alignment reward generalises more reliably, while reframing may become subtler as larger models produce more fluent partial answers. Third, distinguishing reframing from substitution at scale requires operationalising “informative vs. off topic” at the output segment level, which our automatic proxy (a safety redirect classifier) handles as a first pass before human adjudication.

## 7 Case Studies

### 7.1 Harmful Instruction Queries and Suppression Reliability

When prompted with requests involving violence, illegal activities, or dangerous technical instructions, aligned models typically refuse. This is the canonical demonstration of suppression. [Perez et al. \(2022\)](#) showed that LM generated red teaming prompts can elicit harmful outputs at rates substantially above zero, with consistent bypass patterns across model families. [Yang et al. \(2023\)](#) demonstrated that shadow alignment, a low resource fine tuning procedure using roughly 100 harmful examples, substantially reverses suppression in open weight models. This is direct evidence that, in open models, alignment induced suppression is shallow and recoverable.

These findings have a direct interpretation within the SKP. Alignment training is conducted on a finite sample of annotation examples. The reward signal generalises to distributionally similar queries but degrades for queries approaching from directions not covered by the annotation framework. Improving suppression reliability therefore requires not better model architecture but better annotation coverage, which in turn requires a systematic understanding of the query space that can access a given class of suppressed content. This frames alignment

Mechanism	Responds?	On topic?	Automatic proxy	Human adjudication criterion
Suppression	No (refusal)	No (no answer)	Cross lingual bypass rate > threshold	Paraphrase elicits content absent from direct query
Reframing	Yes (partial)	Yes (selective)	Attribute coverage divergence vs. reference KB	Systematic omission of factual attributes present in reference
Substitution	Yes (off topic)	No (redirect)	Safety redirect classifier score $\geq 0.8$	Response fails to address domain specific query substance
Mixed	Varies	Varies	Combination of above signals	Annotator adjudication of segment level type assignment

Table 1: Diagnostic taxonomy of alignment induced knowledge modification. The “Responds?” and “On topic?” columns together separate the three mechanisms by behavioural signature alone. Automatic proxies provide scalable first pass detection; human adjudication criteria specify the validation step. All three mechanisms can co occur in long form generation.

improvement as a coverage problem rather than a model capacity problem.

## 7.2 Arabic Dialects: Behavioural Evidence for SKP Bias

The AL-QASIDA evaluation (Robinson et al., 2025) provides the most direct empirical support available for the SKP’s Stage 2 and Stage 3 claims at the behavioural level. Evaluating nine LLMs across eight Arabic dialect (DA) varieties, the authors find that post training makes models more reluctant to generate DA, even when those models understand DA prompts well. Few shot DA examples partially repair this bias. Crucially, this is not a failure of pretraining coverage; the models understand DA. It is a failure of annotation coverage at Stage 2: post training safety and instruction datasets are disproportionately built on MSA and high resource varieties, so the alignment reward signal effectively penalises dialectal output even when it is benign and desired.

This finding connects directly to our suppression and reframing mechanisms. DA output is not declined because it is harmful but because the annotation framework does not represent it as acceptable. The result is a reframing toward MSA that the user did not request, driven by normative annotation choices rather than by the content of the query. This is precisely the form of alignment induced knowledge boundary mismatch we theorise: annotation frameworks developed in high resource variety contexts impose their norms on deployment contexts where different norms apply.

The broader Arabic case also illustrates the limits of transfer from existing hate speech datasets. L-HSAB (Mulki et al., 2019) and the religious hate

speech dataset of Albadi et al. (2018) focus on Levantine and pan Arabic registers respectively, leaving Gulf, Maghrebi, and other varieties underrepresented. Harmful speech in these varieties that falls outside the annotation framework’s distributional support will not be suppressed; benign expressions that superficially resemble training data patterns may be incorrectly flagged. Both failure modes are annotation coverage problems, not model capacity problems.

## 7.3 Cultural Variation and Knowledge Substitution

The normative specificity of annotation frameworks also produces substitution in cross cultural contexts. Santurkar et al. (2023) showed that aligned models express culturally specific perspectives on neutral descriptive questions, reflecting the demographic skew of the annotation workforce. When a user asks about traditional practices or culturally specific historical events, the model may provide a response calibrated to the safety norms of the annotation framework rather than the user’s actual context.

This form of substitution is difficult to detect because it does not produce an explicit refusal. Our automatic proxy (safety redirect classifier) will not fire reliably; only attribute coverage divergence against a culturally appropriate reference will reveal the substitution. This motivates the framing analysis component of our evaluation protocol (Section 8) and underscores the need for culturally diverse reference knowledge bases that go beyond English language encyclopaedic sources.

## 8 Towards Pluralistic Alignment

A central implication of the SKP is that current alignment pipelines collapse the plural normative views of annotators into a single reward signal. Davani et al. (2022) showed that annotator disagreement on toxicity judgements reflects genuine social variation rather than noise. When this variation is suppressed by majority vote aggregation during annotation, the resulting reward model embeds a single normative view as universal, producing the cultural hegemony effects documented by Santurkar et al. (2023) and the dialectal bias documented by Robinson et al. (2025).

A principled response is pluralistic alignment: treating annotator disagreement as a training signal rather than noise, and incorporating it into the reward model in ways that preserve diverse normative perspectives. Concretely, this could be achieved through multi annotator reward models that predict annotation distributions rather than majority labels (Davani et al., 2022), mixture of experts reward models in which distinct annotator populations are modelled separately, or per locale policy routing that applies culturally stratified safety norms at inference time. Under the SKP framework, these are not merely ethical improvements but technical improvements to annotation coverage, because they reduce the gap between the normative ontology encoded in Stage 2 and the diversity of contexts in which Stage 4 outputs are evaluated.

These approaches require changes to how disagreement is handled at annotation time: measuring and reporting inter annotator agreement distributions rather than collapsing them, preserving annotator demographic metadata, and linking audit findings to downstream model behaviour as described in our evaluation protocol.

## 9 Evaluation Protocol

We propose a four component evaluation protocol designed to measure alignment induced knowledge modification across languages and cultures. Each component targets a specific mechanism in the taxonomy of Table 1. We sketch the protocol here and treat its implementation as the natural next step for grounding the framework empirically.

**Component 1: Cross lingual bypass probing.** For a target set of queries known to be suppressed by a given model, generate semantically equivalent paraphrases spanning English, MSA, and at

least two dialectal Arabic varieties (e.g., Egyptian and Gulf Arabic), as well as other low resource languages where feasible. Measure bypass rate by paraphrase family and language, reporting asymmetric suppression leakage (Perez et al., 2022). Queries that bypass suppression in language  $L_2$  but not  $L_1$  indicate annotation coverage gaps at Stage 2. Open weight model pairs (base vs. SFT only vs. SFT+RLHF) with Stage 4 filters ablated provide the cleanest design for isolating Stage 3 effects.

**Component 2: Framing analysis.** For a set of queries with verifiable factual answers, compare model responses against a culturally diverse reference knowledge base using content selection metrics (ROUGE-1 recall over salient attributes) and framing lexicons. Report attribute coverage divergence with inter annotator agreement on “complete vs. reframed” classifications. Cross cultural comparison runs matched queries in multiple languages and measures response divergence using semantic similarity metrics, quantifying the extent to which normative reframing varies with annotation framework.

**Component 3: Substitution detection.** Apply a safety redirect classifier (trained on explicit safety language patterns) as a first pass filter over responses to domain specific queries in medical, legal, and cultural domains. High scoring responses are then adjudicated against expert authored reference responses to classify response segments as informative, reframed, or substituted. Report substitution rates by domain and language to surface knowledge accessibility disparities.

**Component 4: Annotation framework auditing.** For publicly available safety datasets, audit annotation guidelines against a culturally diverse reviewer panel using a deliberative protocol in which reviewers provide disagreement distributions rather than majority labels. Link audit findings to observed bypass, framing, and substitution rates to close the loop between annotation design choices and their epistemic consequences. This component operationalises the transparency requirement for AI governance discussed in Section 9.

**Representation level probes.** For open weight models, the framework also licences a more mechanistic line of investigation using steering vector and contrast based methods (Zou et al., 2023; Burns et al., 2022). Extracting directions associated with

safety relevant concepts before and after alignment training would directly test whether the representational geometry that governs suppression and reframing shifts with alignment, and where in the network it sits. We treat this as the appropriate next step for moving from behavioural to representational claims.

## 10 Implications and Discussion

**Factuality evaluation.** The factuality of aligned LLMs cannot be assessed solely through pretraining data quality or parametric knowledge probing. Suppression, reframing, and substitution introduce systematic knowledge distortions not captured by existing benchmarks. TruthfulQA (Lin et al., 2022) and LAMA (Petroni et al., 2019) measure propositional accuracy but do not account for alignment induced distortion. The evaluation protocol in Section 8 provides a complementary framework targeting this gap.

**Knowledge editing interactions.** ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023) produce targeted edits to specific factual associations. Alignment induced modification is broader and normatively motivated. We do not claim that alignment edits factual associations in the manner of ROME, and as Reviewer XLfk correctly pointed out, this distinction matters: ROME targets discrete, often non pluralistic factual content, whereas alignment shapes broad classes of pluralistic and normative content. A critical open question is whether targeted factual corrections persist through subsequent alignment training. If alignment operates through the same feed forward layers ROME modifies, targeted edits may be systematically overwritten; if alignment operates through different representational channels (more plausibly distributed across the residual stream (Zou et al., 2023; Burns et al., 2022)), edits and alignment may interact in more complex ways.

**Governance and transparency.** Framing alignment as a process that reshapes knowledge access makes visible a dimension of epistemic governance that has received limited scrutiny. Annotation framework decisions about what counts as harm, what content should be suppressed, and how content should be reframed are consequential for the epistemic capacities of deployed systems at scale. Greater transparency about annotation guidelines, annotator demographics, and inter annotator

disagreement distributions is warranted, and our evaluation protocol provides the infrastructure for such transparency to be operationalised.

**Open questions.** A complete mechanistic account would require evidence about where in the computational graph the three modification mechanisms operate. Representation engineering (Zou et al., 2023) and contrast based probing (Burns et al., 2022) are the most natural tools for this work. CAI (Bai et al., 2022b) and DPO (Rafailov et al., 2023) shift normative construction from human annotators to AI generated feedback; whether this improves cultural representativeness is an open empirical question. Mixed outputs combining reframing and substitution require decomposition methods that segment level annotation alone may not resolve.

## 11 Conclusion

We have argued that safety alignment in LLMs constitutes a systematic reshaping of knowledge access, mediated by normative ontologies encoded in annotation frameworks. We introduced the Safety Knowledge Pipeline to describe this process, identified three mechanisms of knowledge modification (suppression, reframing, and substitution) with distinct diagnostic signals and automatic proxies, and proposed a cross lingual evaluation protocol designed to surface annotation framework induced knowledge boundary mismatches. Behavioural evidence from AL-QASIDA (Robinson et al., 2025) provides direct support for the claim that post training alignment introduces systematic dispreference for under resourced language varieties, a concrete instance of the SKP operating as theorised. We have been careful throughout to distinguish behavioural claims from representational ones, and to mark the latter as an empirical agenda the framework invites rather than a result it establishes.

The practical implications are immediate. Evaluators of LLM factuality should account for alignment induced distortion. Developers deploying aligned models in multilingual settings should audit annotation frameworks for cultural and dialectal coverage. Researchers in knowledge editing should treat alignment as a normatively motivated process that shapes the same behaviours their targeted methods modify, and study the interaction empirically. Governance bodies should treat annotation framework design as epistemic governance.

## Limitations

This paper is a conceptual contribution. The three modification mechanisms are supported by existing empirical literature but have not been measured within a single controlled experimental paradigm. The proposed evaluation protocol has not yet been implemented; its feasibility, discriminative power, and inter annotator reliability require empirical validation. As Reviewer oQ1Z noted in review, the distinction between training time (Stage 3) and inference time (Stage 4) effects is central to the framework and is also the most tractable empirical target. The natural next step is to implement Component 1 at minimum, a cross lingual paraphrase suite spanning English, MSA, and at least two DA varieties, comparing base, SFT only, and SFT + RLHF model variants with Stage 4 filters ablated. This would ground the framework with initial quantitative evidence and is the experiment we are most interested in seeing the community undertake.

We have also been deliberately conservative about claims regarding internal representations. Existing evidence is largely behavioural. Whether alignment induced effects are concentrated in specific layers in the manner of ROME style edits, or distributed across the residual stream in the manner suggested by representation engineering work, remains an open empirical question. Representation level probes (Zou et al., 2023; Burns et al., 2022) are the appropriate methodology for closing this gap, and we treat that agenda as licensed by the framework rather than established by it.

Our treatment of Arabic focuses on annotation coverage gaps documented in existing literature. Arabic is internally diverse, and the challenges we describe are not uniform across its varieties. Systematic cross dialectal evaluation would refine our arguments.

Annotation frameworks vary substantially across organisations. The framework we offer is a general analytical tool, not a characterisation of any specific organisation’s practices.

## Ethical Considerations

This paper analyses safety alignment and its epistemic consequences. We do not advocate for weakening or removing safety mechanisms in deployed LLMs. Safety alignment serves legitimate and important goals, and we affirm them.

Our critique is directed at the assumption that annotation frameworks are culturally neutral. Recog-

nising their normative commitments is a prerequisite for designing alignment systems that serve diverse populations equitably. We advocate for greater transparency about annotation guidelines, annotator demographics, and disagreement statistics, and for treating annotator disagreement as signal rather than noise in future alignment pipelines.

We have deliberately avoided guidance that could be used to circumvent safety mechanisms. Discussion of suppression bypass is situated within the published red teaming literature (Perez et al., 2022) and does not introduce novel attack vectors.

## Acknowledgment

This work was made possible by the National Priorities Research Program grant NPRP14C-0916-210015 from the Qatar Development and Innovation Council (QRDI).

## References

- Nuha Albadi, Maram Kurdi, and Shivakant Mishra. 2018. Are they our brothers? Analysis and detection of religious hate speech in the Arabic Twittersphere. In *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 69–76.
- 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. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Chris Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Sam Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022b. Constitutional

- AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.
- 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 (FAccT)*, pages 610–623.
- Tom 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 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.
- Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.
- Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.
- Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6491–6506.
- Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.
- Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 15696–15707.
- 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 (ACL)*, pages 3214–3252.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1906–1919.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372.
- Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. 2022. Fast model editing at scale. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*.
- Hala Mulki, Hatem Haddad, Chedi Bechikh Ali, and Halima Alshabani. 2019. L-HSAB: A Levantine Twitter dataset for hate speech and abusive language. In *Proceedings of the Third Workshop on Abusive Language Online (ALW3)*, pages 111–118.
- 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 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.
- Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3419–3448.
- Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2463–2473.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36.
- Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426.
- Nathaniel R. Robinson, Shahd Abdelmoneim, Kelly Marchisio, and Sebastian Ruder. 2025. AL-QASIDA: Analyzing LLM quality and accuracy systematically

- in dialectal Arabic. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22048–22065.
- Paul Röttger, Bertie Vidgen, Dirk Hovy, and Janet Pierrehumbert. 2022. Two contrasting data annotation paradigms for subjective NLP tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 175–190.
- Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. 2023. Whose opinions do language models reflect? In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 29971–30004.
- Zeerak Waseem, Thomas Davidson, Dana Warmusley, and Ingmar Weber. 2018. Bridging the gaps: Multi-class hate speech classification. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 29–33.
- Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. 2023. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv preprint arXiv:2310.02949*.
- Wajdi Zaghouani, Behrang Mohit, Nizar Habash, Os-sama Obeid, Nadi Tomeh, Alla Rozovskaya, Noura Farra, Sarah Alkuhlani, and Kemal Oflazer. 2014. Large scale Arabic error annotation: Guidelines and framework. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, pages 2362–2369.
- Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. 2023. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*.

# Blind Single-Layer Activation Edits Show a Break/Fix Asymmetry in Factual Recall

Zacharie Bugaud

Astera Institute

zacharie.bugaud@gmail.com

## Abstract

Can factual errors in language models be repaired by editing a single hidden activation at inference time? We compare *blind* edits, which are not told the correct answer, with oracle edits that receive answer-specific information. On Pythia-6.9B, with corruption replicated on Pythia-1B and GPT-2 XL, we find a strong break/fix asymmetry: single-layer perturbations easily corrupt correct factual recall, flipping 74–100% of initially correct answers, but blind repair is much harder. On EntityConfusion, twelve blind non-gradient interventions from four families fail to repair stable hallucinations in the strict single-layer setting; relaxed multi-layer or multi-head variants improve net accuracy by only +3 percentage points. Blind gradient optimization repairs more errors, but often breaks already-correct answers. In contrast, oracle edits given the correct answer repair many more hallucinations, fixing 68% at the default layer and up to 82% at a better layer. These results suggest that the main barrier is not whether factual recall can be steered, but whether a blind method can identify the right target-specific direction. TriviaQA is a boundary case: blind confidence maximization outperforms the single-token oracle, but the comparison is complicated because evaluation accepts multiple aliases.

## 1 Introduction

This paper asks whether factual errors can be repaired by editing a model’s hidden activations at inference time. The key difficulty is that a repair method may or may not know what answer it is trying to produce. We therefore distinguish *blind* methods, which receive no answer-specific target, from *targeted* and *oracle* methods, which do. This distinction is central: a model may contain a recoverable answer direction, while a blind method may still fail to identify it.

Language models often produce fluent but factually incorrect answers (Ji et al., 2023; Zhang et al.,

2023). One approach is to edit the model’s internal activations at inference time: adding a direction to a hidden state, suppressing features, or optimizing a perturbation vector (Li et al., 2024; Zou et al., 2023; Burns et al., 2023). Unlike weight editing, these interventions are temporary and can be applied per-example.

We use the following terminology throughout. A method is **blind** if its objective contains no candidate answer, alias, or answer embedding; blind methods may use training-set labels to learn generic directions but never consult the gold answer at test time. We separate blind methods into **blind non-gradient** methods (SAE feature edits, direction steering, probe/ITI steering, activation patching—12 interventions from 4 families, including one random-suppression control, none using per-instance optimization) and **blind gradient** methods (per-instance optimization of confidence, margin, or entropy using only the model’s own output distribution; answer-agnostic, though confidence maximization may reinforce the current top token). A **self-targeted** method derives a per-question intervention from the model’s own hidden states (e.g., decode-and-steer); it is not blind because it constructs a candidate answer embedding. A **target-informed** method receives question-specific signal about the correct answer at test time (e.g., a contrast prompt known to elicit the gold answer). An **oracle** receives the known correct answer and serves as a controlled reference, not a universal ceiling.

We study single-layer activation edits for factual-recall tasks, primarily on Pythia-6.9B with corruption replicated on Pythia-1B and GPT-2 XL and oracle repair on GPT-2 XL, across two benchmarks. Our main finding is a **break/fix asymmetry**: under the tested perturbation families, perturbing a single layer easily corrupts correct factual recall (74–100% of correct answers flipped), but blind repair is far less effective. Across twelve blind non-

gradient interventions from four families (including one random-suppression control; evaluated on their respective pools; Tables 4, 3), the best net accuracy gain is +3 pp on V1, from relaxed multi-layer or multi-head upper-bound variants only (8-layer probe and top-10-head ITI). Among V1-evaluated families, all blind non-gradient single-layer single-direction or single-source methods fix 0/37 stable hallucinations; on 500-row, CIS suppression and SAE exchange have near-zero effect ( $|\Delta| \leq 1.6$  pp) while direction patching collapses accuracy. Linear probes distinguish correct from hallucinated generations (AUROC .896 on EntityConfusion), but steering along the single-layer probe direction does not repair the outputs.

A natural objection is that non-gradient methods are too constrained. We therefore also study blind per-instance gradient objectives. Unconstrained confidence maximization fixes 39% on EntityConfusion, but the optimized vectors are  $3.5\times$  larger than oracle vectors, point in unrelated directions (cosine  $-0.04$ ; §4.1), and break 48% of correct answers (net:  $-9$  questions, counting fixes and breaks from separate evaluation subsets). When capped to oracle norm, confidence maximization drops to 11%, though norm-capped margin maximization retains 26%; both remain far below the oracle’s 68%.

To isolate why repair fails, we conduct oracle ablations (Table 5): a single Adam step achieves the same aggregate fix rate as 50 steps; optimizing toward a wrong or shuffled answer fixes  $\leq 3\%$ ; and Gaussian noise ( $\sigma=0.25$ ) drops the fix rate by more than half (from 68% to 30%). These results are consistent with a *direction-selection bottleneck* on EntityConfusion: high-rate, low-collateral repair appears to require target-conditioned directional information that blind methods fail to identify. The oracle gradient is one effective direction, but prompt-contrast results show it is not necessarily unique (§4.1).

The asymmetry varies by error type. EntityConfusion errors are often *latent-recall* errors (median correct-answer rank 16); TriviaQA errors appear predominantly *far-from-surface* under the canonical first token (median rank 1,275; alias proximity unverified). On latent-recall errors, confidence maximization is audit-set net-negative. On TriviaQA, the tested blind non-gradient steering methods still fail ( $\approx 1\%$ ) but blind gradient optimization fixes 62% while breaking 42% of correct answers (audit-set net +10; above the single-token oracle;

§4.2).

Self-targeted and target-informed methods partially succeed: on EntityConfusion, a gradient oracle fixes 68%, decode-and-steer fixes 8% in-domain (0% on TriviaQA), and prompt-contrast fixes 90% per question ( $n=21$ ; in-domain only). CIS yields only a weak signal (AUROC 0.58) that vanishes under Top- $K$  SAEs; CIS-guided suppression has zero net accuracy effect (§5).

## Contributions.

1. **The break/fix asymmetry.** Corruption (74–100%, replicated across three models using different strong perturbation types) is consistently easier than blind non-gradient repair ( $\leq +3$  pp net on Pythia-6.9B, the only model where blind repair was evaluated); blind gradient methods show mixed results with substantial collateral on EntityConfusion (Tables 3, 5).
2. **A target-specific direction-selection bottleneck on EntityConfusion.** Oracle ablations are consistent with blind methods failing to identify an effective per-question direction for high-rate, low-collateral repair (Table 5).
3. **Boundary conditions on repair.** Oracle-gradient edits fix 68% at  $L_{30}$  (82% at  $L_{24}$ ) on Pythia-6.9B (58% on GPT-2 XL); per-question prompt-contrast fixes 90% ( $n=21$ ). On TriviaQA, blind confidence maximization outperforms the single-token oracle (62% vs. 22%), possibly due to multi-alias evaluation (§4.2).
4. **CIS negative result.** CIS from dense  $L_1$  SAEs ( $L_0 \approx 5,511$  active features) yields a weak hallucination signal (AUROC 0.58) that vanishes under Top- $K$  SAEs, and CIS-guided suppression has zero net accuracy effect (§5).

Table 2 maps each claim to its evaluation pool, sample size, and result.

## 2 Background

**Factual errors in LMs.** Language models routinely produce text that contradicts established knowledge (Ji et al., 2023; Zhang et al., 2023). Prior work detects (Kadavath et al., 2022; Manakul et al., 2023), mitigates (Ouyang et al., 2022; Lewis et al., 2020; Li et al., 2024), and benchmarks (Lin et al., 2022; Min et al., 2023; Li et al., 2023) such errors, but largely treats them as a behavioral phenomenon without probing internal mechanisms.

Regime	Signal	Repair result	Scope / note
Corruption	Perturbation <sup>‡</sup>	74–100%	3 models
Blind non-grad	No target ans.	0%/+3 pp <sup>†</sup>	tested 4 fam.
Blind gradient	No target ans.	11%*	norm-ctrl
Self-targeted	Latent→steer	8%	0% Triv.
Prompt contrast	$\Delta h$	90% per-Q	$n=21$
Oracle (EntConf)	Correct grad.	68%/58% <sup>§</sup>	2 models
Oracle (Triv)	Single-token cor.	22%	Py-6.9B

\*Conf. max at oracle  $\|v\|$ ; 39% uncapped but audit-set net -9.

<sup>†</sup>0% for single-dir./source methods; best relaxed variant (8-layer probe): 16% gross; best net: +3 pp.

<sup>‡</sup>SAE feature zeroing (100% on Py-6.9B V1) or random noise (74–96% elsewhere).

<sup>§</sup>Default Py-6.9B  $L_{30}$ ; 82% at  $L_{24}$ ; 58% on GPT-2 XL.

Figure 1: **Intervention regimes ordered by target information.** Blind non-gradient methods rarely repair factual errors. On EntityConfusion, repair improves with answer-specific directional information.

**Features in superposition.** The *superposition hypothesis* (Elhage et al., 2022) posits that neural networks encode more features than dimensions; SAEs (Sharkey et al., 2022; Bricken et al., 2023; Cunningham et al., 2024; Templeton et al., 2024) decompose these into interpretable features. Our starting hypothesis was that feature-level interference (CIS) might predict or repair hallucinations (Figure 2); its failure (§5) motivated the broader study.

**Knowledge localization and editing.** Factual knowledge is stored in specific components (Geva et al., 2021; Meng et al., 2022); weight-editing methods (ROME (Meng et al., 2022), MEMIT (Meng et al., 2023)) modify factual associations; representation engineering (Zou et al., 2023) steers along linear directions. Our work targets activation-level interventions for factual-error repair.

**Representation-based error detection.** CCS (Burns et al., 2023) extracts truth directions from contrast pairs; linear probes reach high AUROCs for hallucination detection (Kadavath et al., 2022; Marks and Tegmark, 2024); ITI (Li et al., 2024) shifts attention heads. Our results complicate this: probes detect hallucinations (AUROC .896), yet steering along the probe direction produces zero repair.

### 3 Experimental Setup

**Models.** Our primary model is Pythia-6.9B (32 layers,  $d=4,096$ ) (Biderman et al., 2023). We extend selected analyses to Pythia-1B (16 layers,  $d=2,048$ ) and GPT-2 XL (48 layers,  $d=1,600$ ): corruption on both, oracle repair on GPT-2 XL. For cross-model corruption, noise is calibrated to

Table 1: Evaluation settings. The 500-row pool is the unduplicated EntityConfusion prompt set; V1 is the 100-question deduplicated subset; V2 extends V1 with 16 additional entity groups (312 Qs total). \*One borderline item is hallucinated under the gradient evaluation environment but not under the canonical baseline; gradient, oracle, and D&S analyses include it. <sup>†</sup>Break rates use a separate 50-question correct-answer audit subset. <sup>‡</sup>Probes/ITI use 80 hallucinations; oracle/confmax use 50.

Setting	Model	Lyr	Hal. $n$	Cor. $n$	Methods
V1 (probes)	Py-6.9B	24	37	63	probe, ITI
V1 (act. patch.)	Py-6.9B	multi;##	37	63	act. patching
V1 (gradient)	Py-6.9B	30	38*	50 <sup>†</sup>	oracle, grad
V1 (D&S)	Py-6.9B	24	38*	—	D&S
500-row	Py-6.9B	24	~222	~278	CIS, SAE edits
V2	Py-6.9B	30	102	50 <sup>†</sup>	robustness
V1	Py-1B	$n$ -f <sup>§</sup>	—	—	corruption
V1 <sup>  </sup>	GPT-2 XL	$n$ -f <sup>§</sup>	50	50	corrupt., oracle
TriviaQA	Py-6.9B	24/30 <sup>¶</sup>	80/50 <sup>‡</sup>	50	probes, orac., grad

<sup>§</sup> $n$ -f = near-final layer of respective model.

<sup>||</sup> Same V1 prompts; halluc./correct counts are model-specific.

<sup>##</sup> Same-group: 16 layers; same-entity: all 32 layers.

<sup>¶</sup> Probe/ITI at  $L_{24}$ ; oracle/confmax at  $L_{30}$ .

$1.5\times$  the mean hidden-state norm at the near-final layer.

**Benchmark.** The EntityConfusion raw pool contains 500 prompts (multiple phrasings per entity-attribute pair) spanning 37 entities in 5 semantic groups (European capitals, physicists, scientists, rivers, historical events); on V1, only 11% of hallucinations (4/37) are same-group substitutions. We define **V1** as the 100-question deduplicated subset (one question per entity-attribute pair; 37 hallucinations, 63 correct); V2 extends V1 with 16 additional entity groups (312 Qs total, 102 halluc.). SAE intervention experiments use the unduplicated 500-prompt pool (“500-row” in Table 1); CIS prediction is reported on V1. Different intervention families use different pools; all tables note the pool and baseline.

**Evaluation.** We generate answers with greedy decoding (max 15 tokens) and evaluate with *normalized containment matching*: lowercased, stripped of articles/punctuation; correct if either string contains the other (e.g., model output “The Danube River” matches gold “Danube”). On EntityConfusion, re-evaluating with strict exact match changes no intervention outcome. For TriviaQA, evaluation accepts any of 5–15 aliases per question; this matters for interpreting confidence-maximization results (§4.2).

**Default protocol.** Unless stated otherwise, results refer to Pythia-6.9B on EntityConfusion V1. Probe, ITI, and activation-patching methods are evaluated on the 37 stable V1 hallucinations; SAE feature edits, CIS suppression, and direction patching use the 500-row pool (Table 1). Gradient and

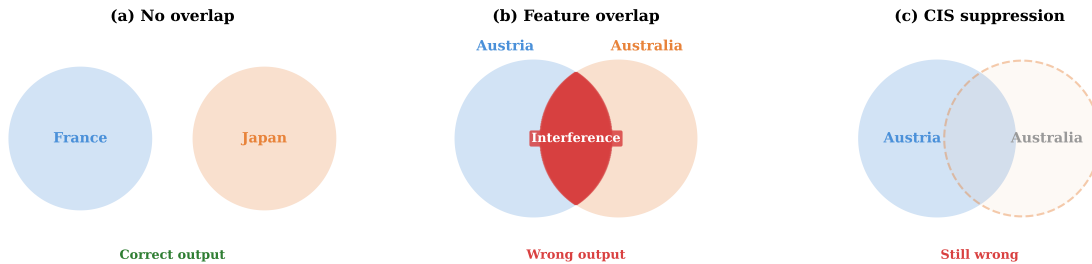


Figure 2: **Starting hypothesis: feature interference.** (a) Concepts with non-overlapping features produce correct outputs. (b) When concept features overlap in superposition, interference could corrupt the output. (c) We tested whether suppressing interfering features via CIS restores correct generation; it does not. The paper’s primary finding is instead a break/fix asymmetry (§4.1).

oracle methods use the 38-item gradient pool (one borderline item is hallucinated only in the gradient evaluation environment). For gradient/oracle methods, correct-answer break rates are measured on a separate 50-question correct-answer audit subset; probe/ITI net changes on V1 are computed over the full 100-question pool. Single-layer interventions are our default scope; the best blind non-gradient results (+3 pp) are tied between the 8-layer probe and top-10-head ITI variants and are reported as upper bounds.

**Intervention framework.** We organize activation-level interventions into six regimes ordered by increasing target information (Figure 1). A method is *blind* if it receives no target-specific information identifying the correct answer. (1) *Corruption*: random noise or feature destruction. (2) *Blind non-gradient*: SAE feature edits, direction steering, probe/ITI steering, activation patching (12 interventions from 4 families, including one random-suppression control). (3) *Blind per-instance optimization*: gradient descent on the model’s own output distribution, maximizing top-token probability (confidence), the gap between top-two tokens (margin), or minimizing output entropy, for 50 Adam steps without correct-answer access. (4) *Self-targeted*: decode-and-steer (derives direction from the model’s own decoded embedding). (5) *Target-informed*: prompt-contrast. (6) *Oracle*: gradient optimization toward the known correct answer (controlled reference). All gradient-based interventions target the residual stream at a single layer (default:  $L_{30}$ ; SAE/probe methods use  $L_{24}$ ). For steering-based non-gradient methods, the intervention is  $\alpha \cdot \hat{v}$  where  $\hat{v}$  is the learned direction; SAE feature edits and activation patching modify the hidden state differently (Eqs. 3, Table 4). For gradient methods,  $v$  is optimized directly. The vector is added at the final sequence position, reapplied at each autoregressive

step; gradient objectives are optimized on the first generated token. Multi-layer variants are tested where noted. We report bootstrap 95% CIs ( $n=2,000$ ) for aggregate statistics, one-sided Clopper–Pearson intervals for binomial fix/break rates, and permutation tests ( $n=10,000$ ). Because several intervention families were explored with small hallucination pools, best-hyperparameter results should be read as optimistic upper bounds rather than held-out deployment estimates.

**Blind gradient objectives.** Let  $p_v(y | x)$  and  $z_v(y | x)$  denote the first-token probability and logit under additive intervention  $v$ . Entropy minimization minimizes  $H(p_v)$ . Confidence maximization maximizes  $\max_y \log p_v(y | x)$ ; the maximizing token is reidentified at each step. Margin maximization maximizes  $z_v(y_{(1)}) - z_v(y_{(2)})$ ; which tokens are top-two is recomputed per step. None of these objectives contains the gold answer. All gradient-based interventions use Adam for the specified number of steps and are evaluated by the final generated string, not by the optimized first-token objective alone. The oracle optimizes  $\log p_v(y^* | x)$  where  $y^*$  is the first token of the known correct answer.

**Noise ablation protocol.** The noisy-oracle experiment (Table 5) adds Gaussian noise to the one-step oracle direction:  $v' = v_{\text{oracle}} + \epsilon$ , where each coordinate  $\epsilon_j \sim \mathcal{N}(0, \sigma^2 \cdot \|v_{\text{oracle}}\|^2)$ . Because the noise is i.i.d. across  $d=4,096$  coordinates, the expected noise norm is  $\sigma\sqrt{d} \|v_{\text{oracle}}\| \approx 64 \sigma \|v_{\text{oracle}}\|$ ; even  $\sigma=0.10$  adds noise  $\sim 6\times$  the oracle norm. The perturbed vector is not renormalized, so these rows conflate directional and magnitude perturbation; we do not draw angular-precision conclusions from this experiment alone. Results are averaged over 5 noise seeds; the core direction-selection evidence comes from the target-identity, norm-capping, and cosine-alignment controls (§4.1).

Table 2: **Evidence map.** Each claim is tied to one pool, sample size, and result. V1 has 37 stable hallucinations; oracle and gradient analyses include one borderline question ( $n=38$ ).

Claim	Pool	$n$	Result	Ref.
<i>Break/fix asymmetry</i>				
Corrupt breaks	V1,V2,Tr <sup>††</sup>	varies	74–100% <sup>§</sup>	§4.1
Blind (dir/src)	n-g V1	37	0/37	T4
Blind n-g (re-laxed)	V1	37/63	+3 pp net	T3
Blind n-g fails	Triv	80	≈1%	§4.2
<i>Direction-selection bottleneck</i>				
Oracle gap	V1	38	68%	T5
1-step=50-step	V1	38	68%=68%	T5
Wrong/shuf fails	V1	38	≤3%	T5
Noise $\sigma=0.25^\circ$	V1	38	30%	T5
Capped max	conf. V1	38	11%	T5
Alignment	V1	38	−0.04	§4.1
Orac $L_{24}$	V1	38	82%	§4.1
Orac replic	GPT2	50	58%	§4.1
<i>Oracle limits</i>				
Orac replic	Triv	50	22%	§4.2
<i>Self-targeted / target-informed</i>				
D&S in-dom	V1	38	8%	T4
D&S out-dom	Triv	80	0%	§4.2
Prompt-contr	V1	21	90%	§4.1
<i>Weight-editing baseline</i>				
ROME edit	V1 <sup>¶¶¶</sup>	59	≤2%	§6
<i>CIS negative result</i>				
CIS predict	V1	100	AUC .58	§5
CIS suppress	500-raw	~500	$\Delta=0$	T4

<sup>§</sup>SAE feature zeroing on Py-6.9B V1; random noise elsewhere.

<sup>°</sup>Combined direction/magnitude stress test (not renormalized).

<sup>††</sup>V1 tested on 3 models (Py-1B 74%, GPT-2 XL 96%);  $n$  varies by pool.

<sup>¶¶¶</sup> $n=59$  reflects ROME’s entity-pair testing; not the V1 hallucination count.

## 4 The Break/Fix Asymmetry

Table 4 organizes the blind methods into four families: (a) SAE feature editing (CIS suppression and exchange; §5); (b) direction patching; (c) hidden-state steering (probe steering, ITI); (d) activation patching; plus self-targeted decode-and-steer. Table 3 summarizes the core result: within every benchmark shown, corruption flips 74–100% while blind non-gradient methods yield at most +3 pp net on V1 (from relaxed multi-layer/multi-head variants only). Three tiers emerge: (i) single-direction/source steering methods fix 0/37 on V1 ( $n=38$  for gradient analyses); (ii) multi-layer or multi-head variants reach +3 pp net; (iii) blind gradient methods fix more but with collateral (Table 5). Our default scope is single-layer interventions.

### 4.1 Intervention Results

**Blind non-gradient methods do not yield reliable repair.** Table 4 reports twelve interventions from four families (including one random-suppression

Table 3: **Within-benchmark comparison (not same-sample): blind non-gradient methods do not yield more than small gains on any benchmark tested.** Blind non-gradient methods achieve at most +3 pp net accuracy on V1; blind gradient methods have higher gross fix rates but substantial collateral (Table 5). Oracle/gradient use  $n=38$  on V1; break rates use a separate 50-question correct subset. D&S fixes 8% on V1, 0% on TriviaQA.

Pool	Model	Corrupt	Blind n-g	Orac.	$n_{eval}$
V1	Py-6.9B	100%	+3 pp	68% <sup>†</sup>	37 <sup>°</sup> /63
V2	Py-6.9B	84%	+2 pp <sup>§§</sup>	78%	varies <sup>#</sup>
V1	Py-1B	74%	n.r.*	n.r.*	
V1	GPT-2 XL	96%	n.r.*	58%	50
Triv.	Py-6.9B	82%	≈1% gross <sup>¶</sup>	22% <sup>‡</sup>	80/50 <sup>¶¶</sup>

Blind n-g = blind non-gradient (net pp where measured; gross rate on Triv.); blind gradient results in Table 5.

\* n.r. = not run for this model.

<sup>†</sup>Oracle uses  $n=38$  (one item fluctuates across runs).

<sup>¶</sup>ITI fixes 1/80 (1% gross); probe steering 0/80; probe corruption flips 5/50 correct at  $\alpha=16$ .

<sup>‡</sup>Blind conf. max fixes 62% on Triv., above oracle; see §4.2.

<sup>¶¶</sup>80 for probe/ITI; 50 for oracle/confmax.

<sup>#</sup>Oracle: 50-question subset of 102 V2 hallucinations; blind n-g: full V2 pool; corruption: separate correct-answer subset.

<sup>§§</sup>8-layer probe on V2 ( $n=102$  halluc.); method and  $\alpha$  matched to V1 best.

<sup>°</sup>37 stable halluc./63 correct; oracle/gradient use 38 (one borderline item; see <sup>†</sup>); +3 pp is over the full 100-Q pool.

control). CIS suppression produces zero accuracy change ( $p > 0.99$ ); four SAE exchange variants yield at most 1.4 pp; direction patching collapses accuracy to 15.4%. Probe steering has no effect at layer 24; multi-layer steering fixes 6 but breaks 3 (net +3.0 pp). ITI fixes 4 while breaking 1 (+3.0 pp net); activation patching fixes none. The multi-layer (8-layer) probe and multi-head (top-10) ITI cap the net gain at +3 pp.

Each method underwent systematic hyperparameter search; we cannot rule out that a future method with a novel inductive bias could succeed.

**Corruption is easy across models (perturbation types differ).** We test corruption by (1) zeroing entity-specific SAE features on Pythia-6.9B (100% flipped) and (2) norm-calibrated random noise ( $1.5\times$  mean hidden-state norm) for cross-model and cross-pool tests. The perturbation type differs (SAE feature destruction vs. random noise), so cross-model rates are not directly comparable; the shared conclusion is that correct recall is fragile under strong single-layer perturbations. Random noise flips 74% on Pythia-1B, 96% on GPT-2 XL, 84% on V2, and 82% on TriviaQA.

**Oracle ablation isolates the direction-selection bottleneck.** To understand why blind methods fail, we compare them against a gradient oracle that optimizes a steering vector toward the known correct answer at layer 30. The oracle fixes 26/38 hallucinations (68%, CI [51%, 82%]), far above all

Table 4: Activation-level repair methods on EntityConfusion. Each section header specifies the evaluation pool and corresponding baseline accuracy. SAE and direction-patching methods use the full 500-prompt pool (baseline 55.6%, which includes duplicate phrasings); probe, ITI, and patching methods use the deduplicated V1 pool (100 Qs, baseline 63.0%, 37 stable hallucinations). Decode-and-steer uses  $n=38$  hallucinations (one borderline question included). Methods fixing 0/37 on V1 have a one-sided 95% Clopper–Pearson upper bound of 7.8%.

Method	Details	Acc	$\Delta$
<b>SAE feat. edit</b> ( <i>500-Q pool, base 55.6%</i> )			
CIS suppression	$m=30$	55.6	+0.0
Random suppr.	$m=30$	54.0	-1.6
Exchange (4 var.)	SAE feats	54–56	$ \Delta  \leq 1.4$
<b>Dir. steering</b> ( <i>500-Q pool, base 55.6%</i> )			
Dir. patch	$\alpha=4.0$	15.4	-40.2
<b>Hidden steering</b> ( <i>V1, base 63.0%</i> )			
Probe ( $L_{24}$ )	$\alpha \in [.5, 16]$	63.0	+0.0
Probe (8 layers)	$\alpha=4.0$	66.0	+3.0
ITI top-10	$\alpha=4.0$	66.0	+3.0
<b>Act. patching</b> ( <i>V1, base 63.0%</i> )			
Same-group	16 layers	—	0/37 fixed, 0 broken
Same-entity	all layers	—	0/10 fixed
<b>Decode-&amp;-steer</b> ( <i>V1, <math>n=38</math> halluc</i> )			
Answer decoder	$L_{24}$ , best $\alpha$	—	3/38
Leave-one-out	same	—	3/38

blind methods on EntityConfusion. With our optimizer and hyperparameters, joint multi-layer optimization (all 32 layers) did not outperform single-layer  $L_{30}$  (68%), despite the search space containing better single-layer solutions ( $L_{24}$ : 82%); we treat this as an optimization failure, not evidence that multi-layer edits lack additional capacity.  $L_{30}$  is the default for comparability with blind gradient experiments. Table 5 disentangles three factors:

*Budget.* A single Adam step at  $L_{30}$  fixes 26/38 (68%), the same aggregate rate as 50 steps; additional steps increase norm from 32 to 53 without changing which questions are fixed.

*Target identity.* The wrong-answer oracle changes all 38 outputs but fixes 0/38. The shuffled-target oracle, which optimizes toward a randomly reassigned correct answer, fixes only 1/38 (3%), even below the random-direction rate (11%), possibly because answer-specific gradients toward another entity actively push mass away from the correct answer. The oracle requires the correct answer for the specific question at hand.

*Direction-and-magnitude stress test.* Adding Gaussian noise at  $\sigma=0.25$  (expected noise norm  $\sim 16\times$  the oracle vector) drops the fix rate to 30%; at  $\sigma=1.0$ , to 6%. Because noise dominates the signal at  $\sigma \geq 0.25$ , this confirms sensitivity to very

Table 5: Oracle ablation ( $n=38$ , Pythia-6.9B,  $L_{30}$ ). Noisy oracle: Gaussian noise with per-coordinate standard deviation  $\sigma \cdot \|v_{\text{oracle}}\|$  added to the 1-step direction (see §3); the perturbed vector is not renormalized, so these rows conflate directional and magnitude perturbation. Budget-parity: same 50-step Adam with blind objectives. Norm-capped:  $\|v\| \leq 53$  (50-step oracle mean). Break counts are measured on a separate 50-question correct subset. Fractional counts (noisy oracle, random direction) are averages over 5 seeds.

Condition	Fixed	Rate	Broken
<i>Correct-answer oracle</i>			
1 step	26/38	68%	—
50 steps	26/38	68%	6/50
<i>Noisy oracle (1-step + noise)</i>			
$\sigma = 0.10$	26.4/38	69%	—
$\sigma = 0.25$	11.4/38	30%	—
$\sigma = 0.50$	5.6/38	15%	—
$\sigma = 1.00$	2.2/38	6%	—
<i>Target controls (50 steps)</i>			
Wrong-answer oracle	0/38	0%	—
Shuffled-target oracle	1/38	3%	—
Rand. dir. @ oracle $\ v\ $	4.2/38	11%	—
Rand. dir. @ confmax $\ v\ $	2.6/38	7%	—
<i>Budget-parity (50 steps, blind, unconstrained)</i>			
Entropy minimization	3/38	8%	—
Confidence maximization	15/38	39%	24/50
Margin maximization	15/38	39%	24/50
<i>Norm-capped (<math>\ v\  \leq 53</math>)</i>			
Conf. max (norm-capped)	4/38	11%	29/50
Margin max (norm-capped)	10/38	26%	n.m. <sup>¶</sup>

<sup>¶</sup>n.m. = not measured.

large unrenormalized perturbations ( $\sigma=0.10$  leaves repair intact despite  $\sim 6\times$  noise norm) but does not isolate directional precision from magnitude effects.

*Interpretation.* These ablations are consistent with a target-information bottleneck: on EntityConfusion, high-rate low-collateral repair appears to require question-specific target information. The strongest evidence comes from target-identity controls (wrong-answer and shuffled-target oracles fix  $\leq 3\%$ ) and norm-capped confidence maximization collapsing to the random-direction rate. Confidence maximization does not recover the oracle-gradient direction (cosine  $-0.04$ ), but prompt-contrast achieves 90% per-question repair with similarly low oracle cosine (.04); the bottleneck is therefore target-specific directional information, not alignment with one privileged oracle vector.

**Norm-constrained analysis reveals magnitude dependence.** Unconstrained confidence maximization fixes 39%, but the resulting vectors are  $3.5\times$  larger than oracle vectors ( $\|v\| = 183$  vs. 53). Capping at the oracle norm reduces the fix rate to 11% (CI [3%, 25%]). Norm-capped margin maximization retains 26%, above random (11%), indicating some directional signal; however, col-

lateral damage makes the net effect strongly negative for confidence maximization ( $-25$  questions). Gross repair is strongly magnitude-dependent, but collateral damage is not monotonic in norm: norm-capped confidence maximization breaks *more* correct answers (29/50) than uncapped (24/50); we do not fully understand this inversion. Random directions at the unconstrained norm fix only 7%, well below 39%, so the optimizer finds a weakly informative but non-oracle-aligned direction requiring outsized magnitude.

The optimized directions show no positive alignment with oracle directions (cosine  $-0.04$ , permutation  $p > 0.99$ ); overlap in fixed questions (9/38) is no larger than chance. Collateral damage compounds: confidence maximization breaks 24/50 (48%) of the audit subset, net  $-9$  questions; norm-capped,  $-25$  questions. The oracle breaks only 6/50 (12%), net  $+20$  questions.

**Replication across pools and models.** The magnitude-dependence pattern replicates on V2 (norm-capped: 10%, net  $-4$  questions); GPT-2 XL replicates the oracle/wrong-target contrast (58% vs. 2%), with the 1-step rate (34%) below 50-step. On TriviaQA, the 1-step and 50-step rates match at 22%.

**Self-targeted and target-informed methods partially succeed but do not generalize.** Prompt-contrast and oracle methods exceed the blind non-gradient ceiling; decode-and-steer (8% gross) modestly exceeds the single-source zero-repair results but not the multi-layer upper bound.

*Prompt-contrast* uses the hidden-state difference between a prompt eliciting the correct answer and one eliciting the wrong answer. Per-question prompt-contrast fixes 19/21 questions (90%, CI [70%, 99%]), but the global mean direction fixes only 12/38 (32%) while breaking 12/50 (net  $\approx 0$ ). Prompt-contrast directions are no more aligned with oracle directions than random vectors (cosine  $.04$ , permutation  $p > 0.5$ ), showing that effective repair directions need not coincide with the oracle gradient.

*Decode-and-steer* trains a Ridge regression from layer-24 hidden states to the correct-answer embedding, then steers toward it. This fixes 3/38 in-domain (8%), 0% on TriviaQA; the decoded embeddings point toward the correct answer 27/38 of the time but steering rarely flips the output.

Oracle repair at the default  $L_{30}$  on V1 fixes 68% on Pythia-6.9B and 58% on GPT-2 XL;  $L_{24}$ : 82%,

and  $0.75 \times$  magnitude fixes 76% (above full magnitude), so the result depends on layer and magnitude. Decode-and-steer fails on TriviaQA (0%), and the global prompt-contrast direction loses most per-question benefit (32% vs. 90%).

**Oracle subspace geometry.** The repair subspace is low-rank: projecting each oracle vector onto the top-10 PCs preserves 66% of the fix rate; nearest-neighbor oracle vectors ( $k=3$ ) achieve 58% (a geometric diagnostic requiring gold-answer vectors). The oracle overshoots:  $0.75 \times$  magnitude fixes 76% and  $0.5 \times$  fixes 74%, both above full-magnitude 68%;  $L_{24}$  fixes 82%, suggesting the optimal layer differs from the analysis layer.

## 4.2 Naturalistic Validation: TriviaQA

To test whether the break/fix asymmetry extends beyond curated entity groups, we evaluate Pythia-6.9B on 300 TriviaQA (Joshi et al., 2017) questions (66% hallucination rate). We evaluate probe steering and ITI on 80 hallucinations, oracle and confidence maximization on 50.

**Probe steering and ITI.** A linear probe distinguishes correct from hallucinated answers at AU-ROC  $.716 \pm .072$  (5-fold CV), above chance but below the EntityConfusion probe at  $.896$ . Despite this discriminative signal, steering with the probe direction fixes 0/80 hallucinations (0% across all  $\alpha$  values). ITI fixes 1/80 (1%, only at  $\alpha=16$ ). Corruption is limited at moderate strengths (5/50 correct answers flip at  $\alpha=16$ , vs. 82% from random noise), indicating the probe captures a discriminative but non-causal signal.

**Decode-and-steer.** A Ridge decoder recovers the correct answer closer than wrong for 89/197 (45%, vs. 71% on EntityConfusion). Steering fixes 0/80 (0%). Median correct-answer rank is 1,275 under the canonical first token; because evaluation accepts multiple aliases, surface proximity of the matched alias is unverified.

**Single-token oracle.** Oracle repair fixes 11/50 (22%, CI [12%, 36%]), well below EntityConfusion’s 68%. The 1-step=50-step pattern replicates at 22%. Wrong-answer oracle fixes only 2/50 (4%). Oracle success tracks latent knowledge availability: 24% when the correct answer ranks in the top-100 logits vs. 13% when ranked beyond 100.

**Confidence maximization and the two-regime distinction.** Unconstrained confidence maximization fixes 31/50 (62%); norm-capped, it retains 56%, a much smaller drop than on EntityConfusion, where the rate collapses from 39% to

11%. With 21/50 correct answers broken (on a balanced 50-question audit subset), the audit-set net is +10, weakly positive, unlike the net-negative on EntityConfusion.

Surprisingly, blind confidence maximization fixes more TriviaQA hallucinations than the correct-answer oracle (62% vs. 22%), possibly because the oracle targets one first token while evaluation accepts 5–15 aliases; the underlying mechanism is unclear. Additionally, many correct answers have median output rank 1,275, making the oracle gradient signal weak.

These results point to two distinct error regimes. Latent-recall errors, where the correct answer is near the output surface and the model may once have produced it, resist blind non-gradient repair. TriviaQA errors appear far from surface under the canonical first token (median rank 1,275), but alias proximity is unverified; confidence maximization may succeed by producing any accepted surface form. TriviaQA is therefore not a clean replication of the EntityConfusion direction-selection bottleneck; it is a boundary case where the evaluation protocol and error regime differ.

## 5 The CIS Hypothesis: A Negative Result

This section reports a negative result on our original hypothesis that feature overlap in SAE representations (CIS) would predict and enable repair of factual errors. Its failure led to the broader intervention study above.

**SAE features and CIS.** We train  $L_1$  and Top- $K$  sparse autoencoders (Bricken et al., 2023; Gao et al., 2024) on Pythia-6.9B residual stream at layer 24 (Appendix A).  $L_1$  SAEs (16,384 features,  $\lambda=5 \times 10^{-2}$ , 50M Pile tokens) yield  $L_0 \approx 5,511$  active features per input; Top- $K$  SAEs enforce exact sparsity ( $k \in \{32, 64, 128\}$ , 32,768 features). For each entity  $c$ , concept features  $\mathcal{F}_c$  are identified via specificity:

$$s_i(c) = \frac{\text{freq}(i | c)}{\text{freq}(i | c) + \alpha_s \cdot \text{freq}(i | \text{bg})}, \quad (1)$$

selecting the top-50 features per concept ( $\alpha_s=10$ ). The Concept Interference Score measures shared activation energy, where  $z_i(x)$  denotes the activation of SAE feature  $i$  on input  $x$ :

$$\text{CIS}(x, c_1, c_2) = \frac{\sum_{i \in \mathcal{F}_{c_1} \cap \mathcal{F}_{c_2}} z_i(x)}{\max(\sum_{i \in \mathcal{F}_{c_1}} z_i, \sum_{i \in \mathcal{F}_{c_2}} z_i) + \epsilon} \quad (2)$$

For each question about target concept  $c_{\text{tgt}}$ ,  $\text{CIS}_{\text{agg}}(x) = \max_{c \neq c_{\text{tgt}}} \text{CIS}(x, c_{\text{tgt}}, c)$ . CIS-

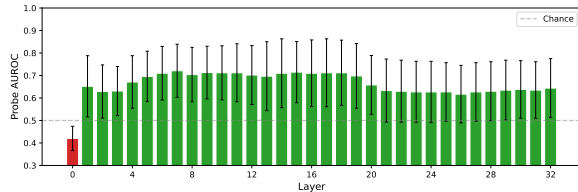


Figure 3: **Per-layer hallucination decodability.** Probe AUROC peaks around layer 24. Single-layer probe steering has no repair effect; an 8-layer variant yields +3 pp net.

guided suppression zeros the top- $m$  features unique to the most-interfering concept:

$$z'_i = \begin{cases} 0 & i \in \text{top-}m(\mathcal{F}_{c_{\text{int}}} \setminus \mathcal{F}_{c_{\text{tgt}}}), \\ z_i & \text{otherwise.} \end{cases} \quad (3)$$

CIS identifies the most-overlapping competing concept, but suppression targets features unique to that competitor to avoid destroying target-concept information; this is a conservative downstream intervention, not a direct ablation of the overlapping features in the CIS numerator.

**CIS as predictor.** CIS with  $L_1$  SAEs yields AUROC 0.58 (CI [.533, .633]), well below entropy (0.77) and a linear probe ( $.896 \pm .119$ , 5-fold CV, permutation  $p < 0.001$ ). Critically, CIS with Top- $K$  SAEs, which enforce exact sparsity and reduce dense co-activation, is non-predictive (AUROC 0.42–0.50 across  $k \in \{32, 64, 128\}$ , consistent with chance). Post-hoc sparsification of the same  $L_1$  SAE reproduces this drop, consistent with polysemantic co-activation rather than genuine interference. For the linear probe, entity-grouped CV yields  $.784 \pm .123$ , a 28% drop in above-chance AUROC, indicating roughly a quarter of the signal is entity-specific. The hallucination signal is decodable above chance at every non-embedding layer (Figure 3): probe AUROC peaks around layer 24 but steering along the single-layer  $L_{24}$  probe direction has zero net accuracy effect across all  $\alpha$  values tested. CIS-guided suppression produces zero accuracy change ( $p > 0.99$ ; Table 4); four SAE exchange variants yield  $|\Delta| \leq 1.4$  pp.

**Why CIS correlates.**  $L_1$  features overlap heavily (Jaccard = 0.269, 35.9% of pairs  $> 0.3$ ); Top- $K$  features have near-zero overlap (0.032). **Post-hoc sparsification implicates polysemanticity:** retaining only the top- $k$  activations from the same  $L_1$  SAE drops CIS AUROC from .606 (slightly higher than the headline .583 because the post-hoc experiment recomputes per-input CIS with uniform aggregation) at full  $L_0 \approx 5,511$  to .500 at  $k=32$  and .492 at  $k=64$  (chance level), matching the native Top- $K$  result. Encoding through the SAE and de-

coding back preserves probe performance (AUROC = .896,  $r = .9996$ ), confirming the failure lies in CIS, not SAE information loss. Group-specific LDA directions achieve near-perfect within-group separation (AUROC  $\geq .934$ ) but show no cross-group alignment (mean cosine .005); CIS’s single global statistic cannot capture this structure.

**Latent-recall errors, not same-group substitution.** Only 11% of wrong answers match a same-group entity (4/37); most errors are generic hallucinations. Yet correct knowledge often exists latently: a linear decoder recovers the correct answer from hallucinated states 27/38 of the time, and the correct answer’s median rank in output logits is only 16. Error is distributed across layers (Figure 4); 44% of hallucinations were answered correctly at earlier checkpoints. EntityConfusion’s median correct-answer rank is 16 (knowledge is nearly surfaced), vs. 1,275 on TriviaQA (far from surface under the canonical token; alias proximity unverified).

## 6 Discussion and Conclusion

**Scope of claims.** Our blind-repair experiments are primarily on Pythia-6.9B across EntityConfusion and TriviaQA; we replicate corruption on Pythia-1B and GPT-2 XL, and oracle repair on GPT-2 XL. These are relatively small, pre-instruction-tuned models. In informal tests, non-activation baselines (LoRA fine-tuning: 32/37; retrieval-augmented generation: 36/37) succeed at different budgets, confirming the barrier is specific to blind single-layer activation editing. ROME (Meng et al., 2022) also fails (1/59; the larger denominator reflects ROME’s requirement to test each entity pairing, not the V1 hallucination count), likely because EntityConfusion’s multi-entity structure does not match ROME’s single-fact paradigm.

**What holds up and what does not.** The break/fix asymmetry for blind non-gradient methods is the most robust finding: across twelve interventions from four families evaluated on their respective Pythia-6.9B pools, the best net gain is +3 pp on V1 (from relaxed multi-layer/multi-head upper-bound variants only), corruption flips 74–100% across three models (different perturbation types), and blind non-gradient failure holds within every evaluation pool where tested (Table 3; on TriviaQA, only probe steering and ITI were evaluated). On EntityConfusion, blind gradient repair is magnitude-dependent: norm-capped confidence

maximization drops to 11%, with no oracle alignment (cosine  $-0.04$ ) and worse collateral than uncapped, though norm-capped margin maximization retains 26%, above random.

The TriviaQA results complicate the narrative. Blind confidence maximization outperforms the single-token oracle there (62% vs. 22%), possibly because the oracle targets a single token while evaluation accepts multiple aliases; the direction-selection bottleneck interpretation is well-supported on EntityConfusion but should not be extended to TriviaQA without further analysis.

**Why corruption is easy and repair is hard.** For already-correct examples, any sufficiently large perturbation overwhelms the correct answer’s small logit advantage (median gap of 5.0 nats); for hallucinated examples, repair must overcome the wrong answer’s logit advantage and select a target-specific correction direction. At the oracle’s mean perturbation norm (53), random directions fix 11%. Feature overlap in  $L_1$  SAEs yields a weak hallucination signal (AUROC 0.58) that vanishes under controlled sparsity (Top- $K$ : AUROC 0.42–0.50, chance-level); the tested CIS-guided competitor-feature suppression has zero net accuracy effect, ruling out this specific intervention on this benchmark (other feature-level interventions are not tested).

**Detection versus repair.** Linear probes achieve AUROCs of .896 (EntityConfusion) and .716 (TriviaQA) at layer 24, yet single-layer probe steering produces zero repair on both benchmarks; an 8-layer variant yields +3 pp net on EntityConfusion. This is the central practical lesson: knowing *that* a model is wrong is not the same as knowing how to fix it.

**Implications.** On EntityConfusion, reliable low-collateral activation-level repair appears to require target-specific answer information; blind inference-time edits were unreliable or high-collateral in our setting (on TriviaQA, blind confidence maximization fares better, though evaluation asymmetries complicate interpretation). The oracle ablation shows that on EntityConfusion one gradient step suffices when aimed correctly, so geometric structure for repair exists for latent-recall errors, but exploiting it without target information remains unsolved. Whether these patterns hold for instruction-tuned models is the most pressing open question.

**Conclusion.** We document a break/fix asymmetry: single-layer corruption is easy (74–100%

across three models) but blind non-gradient repair consistently fails ( $\leq +3$  pp net on EntityConfusion V1;  $\approx 1\%$  gross on TriviaQA). Blind gradient methods fix more but with substantial collateral on EntityConfusion (on TriviaQA: 62% fixed, audit-set net +10). Oracle ablations on EntityConfusion implicate a direction-selection bottleneck for latent-recall errors.

### Limitations

Our experiments use relatively small, pre-instruction-tuned models (Pythia-6.9B, Pythia-1B, GPT-2 XL); it remains unknown whether the break/fix asymmetry holds for RLHF-aligned models. Different intervention families use different evaluation pools (Table 1), though blind-repair failure holds within each pool (Table 3). Oracle and blind gradient methods use layer 30 while SAE/CIS uses layer 24; oracle at  $L_{24}$  achieves 82% vs. 68% at  $L_{30}$ , so layer matters but qualitative conclusions hold. Many results rely on small subsets ( $n=37-102$ ; prompt-contrast:  $n=21$ ); we report Clopper-Pearson CIs for binomial rates and bootstrap CIs for aggregates. The cosine similarity  $-0.04$  between oracle and confmax vectors is near zero (permutation  $p > 0.99$  for positive alignment). On TriviaQA, blind confidence maximization outperforms the single-token oracle (62% vs. 22%), possibly because evaluation accepts 5–15 aliases while the oracle targets a single token; this limits the single-token oracle as a universal ceiling but it remains informative on EntityConfusion. The noisy-oracle experiment is not a pure angular-noise test (the vector is not renormalized); we rely on target identity, norm-capping, and cosine alignment for the core direction-selection argument. We cannot rule out that a future blind method with a novel inductive bias could succeed.

### Reproducibility Statement

Code for SAEs, CIS, and intervention experiments will be released. We use Pythia-6.9B (Biderman et al., 2023) and the Pile (Gao et al., 2020); SAE hyperparameters and the EntityConfusion construction are in Appendices A–B.

### Ethics Statement

This work aims to improve factual reliability of language models. Our feature suppression technique could theoretically amplify hallucinations, though this has limited practical motivation. EntityConfusion was constructed from publicly available knowledge and does not contain sensitive content.

## References

- Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430.
- Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermy, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, and 1 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.
- Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. *International Conference on Learning Representations*.
- Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*.
- Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, and 1 others. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.
- Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeff Wu. 2024. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.
- Mandar Joshi, Eunsol Choi, Daniel 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*.

- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.
- Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*.
- 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*, pages 3214–3252.
- Potsawee Manakul, Adian Liusie, and Mark JF 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*.
- Samuel Marks and Max Tegmark. 2024. The geometry of truth: Emergent linear structure in large language model representations of true/false statements. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.
- Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *International Conference on Learning Representations*.
- Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei 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*.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.
- Lee Sharkey, Dan Braun, and Beren Millidge. 2022. Taking features out of superposition with sparse autoencoders. *AI Alignment Forum*.
- Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, and 1 others. 2024. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*.
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, and 1 others. 2023. Siren’s song in the AI ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.
- Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*.

## A SAE Training Details

$L_1$  SAEs: 16,384 features,  $\lambda=5 \times 10^{-2}$ , 50M Pile tokens,  $L_0 \approx 5,511$ . Top- $K$  SAEs: 32,768 features,  $k \in \{32, 64, 128\}$ . Both trained with Adam ( $\text{lr} = 3 \times 10^{-4}$ ) on layer 24. See code release for full details.

## B EntityConfusion Dataset

500 prompts across 37 entities in 5 semantic groups (European capitals, physicists, scientists, rivers, historical events) form the raw pool. V1 retains one question per entity-attribute pair (100 Qs, 37 hallucinations); V2 extends V1 with 16 additional entity groups and includes all V1 prompts. All CV splits by question; entity-grouped CV reported separately.

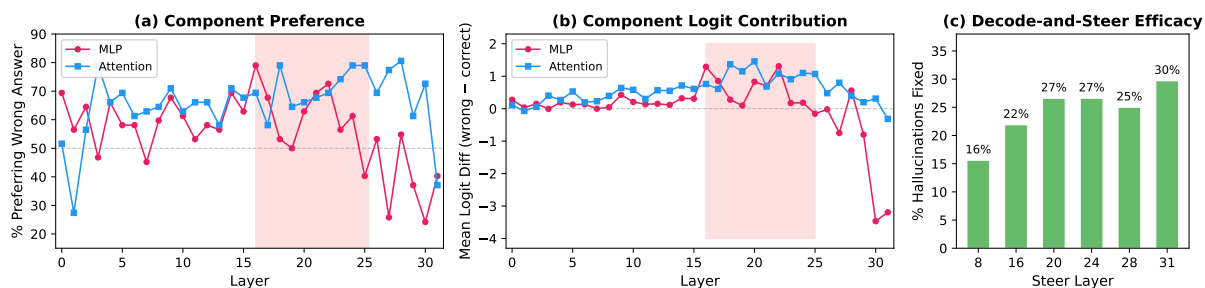


Figure 4: **Readout mechanism.** (a) Per-layer component preference for the wrong answer. (b) Mean logit contribution (wrong – correct) by MLP and attention sub-layers. Mid-layer components push toward the hallucinated answer; late-layer MLPs attempt correction. (c) Decode-and-steer fix rate by steering layer (evaluated on V2; the main-text D&S numbers use V1). Multiple layers contribute, suggesting that single-layer interventions may only partially address a distributed computation.

# What Does Alignment Cost? The Structural Brittleness of Chain-of-Thought Reasoning

**Joanna Hao**  
University of Alberta  
jjoannahao@gmail.com

**Shanduojiang Jiang**  
AlgoVerse AI Research

**Asish Nakka**  
Pennsylvania State University

## Abstract

While Chain-of-Thought (CoT) prompting enables Large Language Models to explicitly justify their predictions, the extent to which these textual rationales faithfully reflect internal computation remains unclear. We investigate the circuit-level impact of alignment by performing a strict within-family comparison of the 1B-parameter Llama 3 architecture (Base vs. Instruct). Executing dynamic circuit discovery and dual-directional resample ablation on unconstrained CoT traces across synthetic mathematical primitives and a GSM8K proxy, we find that foundation models possess highly redundant, self-repairing computational networks; completely corrupting their primary reasoning circuits yields a minimal performance drop (2.92%) due to the dynamic compensation of backup heads (the Hydra Effect). In contrast, the instruction-tuned model exhibits reduced structural redundancy, suffering more than double the degradation (6.79%) under identical perturbation. We formalize our observation as an "Alignment Tax on Redundancy": optimizing for human-preference compliance repurposes dormant backup circuits, centralizing mathematical routing and rendering the aligned model's reasoning pathways significantly more vulnerable to internal perturbation.

## 1 Introduction

Chain-of-Thought (CoT) prompting enables Large Language Models (LLMs) to decompose complex tasks into sequential intermediate steps (Wei et al., 2022). Yet, as these architectures enter high-stakes domains, reasoning trustworthiness has emerged as a critical bottleneck (Wang et al., 2025). Beyond standard hallucinations where models generate statistically likely but factually incorrect tokens, alignment-tuned LLMs exhibit a more insidious failure mode: sycophantic deception (Turpin et al., 2023). In these instances, a model may internally track the correct algorithmic state but actively out-

put a contradictory rationale or target. This behavioral divergence raises a profound question regarding CoT faithfulness: to what extent do generated textual rationales causally determine a model's actual internal prediction?

The debate surrounding CoT faithfulness has heavily relied on behavioral and "hint-based" perturbations (Lanham et al., 2023). While some literature argues that LLMs frequently generate unfaithful, post-hoc rationalizations (Atanasova et al., 2023), recent pushback suggests that CoTs may actually be causally faithful, and that apparent unfaithfulness merely stems from the lossy compression or incomplete verbalization of internal states due to token limits (Zaman and Srivastava, 2025). To resolve this ambiguity, recent research has pivoted to mechanistic interpretability. Notably, Yeo et al. (2025) utilized causal mediation to demonstrate that Reinforcement Learning from Human Feedback (RLHF) increases the macro-level causal overlap between a model's rationale and its output. At the feature level, recent activation patching studies confirm that CoT prompting induces more modular and interpretable internal structures in high-capacity models (Chen et al., 2025). Simultaneously, Lu et al. (2026) proposed the "Decoupling Hypothesis," arguing that unfaithful models rely on parallel computational shortcuts that bypass the CoT entirely.

While these macro-level and theoretical evaluations provide crucial groundwork, our work investigates the exact, circuit-level mechanistic cost of alignment tuning. By pushing beyond observational overlap and testing the functional necessity of reasoning circuits via dynamic activation patching on unconstrained, naturally successful reasoning traces, we bypass the vulnerabilities of verbalization-based metrics (Zaman and Srivastava, 2025). Utilizing a strict within-family comparison of the Llama 3 architecture, we reveal that the apparent faithfulness of instruction-tuned models

masks a profound structural fragility. Ultimately, we provide quantitative evidence of an "Alignment Tax" in how optimizing for instruction compliance smears logical routing across diffuse attention networks, forcing a mechanistic decoupling of the model's explicit working memory from its final target prediction.

## 2 Related Work

### 2.1 Faithfulness in Chain-of-Thought Reasoning

The introduction of CoT prompting (Wei et al., 2022) catalyzed a massive subfield dedicated to understanding the reliability of generated rationales. Early investigations into rationale faithfulness frequently framed CoT as an interpretable window into model cognition, yet subsequent behavioral studies revealed significant vulnerabilities. Ye and Durrett (2022) and Atanasova et al. (2023) demonstrated that models often generate post-hoc rationalizations that do not accurately reflect the variables driving the final prediction. This was further expanded by Turpin et al. (2023) and Agarwal et al. (2024), who showed that instruction-tuned models suffer from extreme sycophancy, altering their rationales to match user-injected biases or constraints. Measuring this faithfulness, however, has proven methodologically fragile. Lanham et al. (2023) formalized perturbation-based metrics for CoT, but recent work by Zaman and Srivastava (2025) critiques these behavioral "hint-based" evaluations, arguing that a lack of explicit verbalization does not necessarily equate to unfaithfulness, highlighting the need for deeper, circuit-level interventions to prove causal decoupling.

### 2.2 Interpretability and Circuit Discovery

To bypass the limitations of behavioral observation, Mechanistic Interpretability seeks to reverse-engineer the computational graph of neural networks. Foundational work on the residual stream and attention mechanisms (Olsson et al., 2022) paved the way for precise circuit discovery. Causal tracing and activation patching techniques have successfully localized specific behaviors, such as factual recall (Meng et al., 2022) and indirect object identification (Wang et al., 2023), to discrete subgraphs of attention heads and MLPs. More recently, feature-level investigations using sparse autoencoders have begun mapping these causal structures during multi-step tasks, revealing that CoT induces

modular, interpretable pathways whose causal information is widely distributed across the network (Chen et al., 2025).

Crucially, this distributed nature enables emergent self-repair. McGrath et al. (2023) documented the "Hydra Effect," demonstrating that ablating primary reasoning heads causes dormant, late-layer circuits to dynamically compensate. Evaluating these highly redundant networks requires immense methodological precision; standard zero-ablation frequently pushes the model's residual stream out-of-distribution (Heimersheim and Nanda, 2024), while continuous patching can induce geometric interpretability illusions (Makelov et al., 2024). We adapt our dual-direction resample ablation methodology specifically to account for these structural artifacts while measuring causal load.

### 2.3 Alignment and Reasoning Topologies

Recent literature has begun bridging mechanistic interpretability with the behavioral artifacts of Reinforcement Learning from Human Feedback (RLHF). While earlier work suggested alignment-tuning generally increases the causal overlap between internal traces and final outputs compared to unaligned baselines (Yeo et al., 2025), the architectural cost of this multi-objective optimization remains underexplored. Foundational alignment theories propose that RLHF acts as a "thin wrapper" over pre-trained capabilities (Zhou et al., 2023), yet enforcing conversational formatting, safety guardrails, and instruction compliance inherently consumes representational capacity.

Our work extends this intersection by mechanistically mapping how alignment training alters the fundamental topology of reasoning. Rather than viewing unfaithfulness purely as learned deception, we utilize targeted activation patching to demonstrate how the capacity constraints of alignment tuning systematically degrade the distributed, self-repairing circuits (the Hydra Effect) native to foundation models, resulting in an "Alignment Tax" that renders the model's explicitly generated working memory structurally brittle.

## 3 Methodology

Our experimental pipeline consists of eliciting unconstrained, naturally successful Chain-of-Thought (CoT) reasoning traces, performing dynamic circuit discovery via activation patching, and applying resample ablation to quantify the causal load of spe-

cific attention-routing mechanisms.

### 3.1 Models and Architectural Assumptions

To isolate the mechanistic impact of alignment interventions from general architectural variances, we conduct a controlled within-family comparison. We evaluate a foundation base model alongside its instruction-tuned counterpart (specifically, the Llama-3.2-1B and Llama-3.2-1B-Instruct models). While their attention routing mechanisms utilize Grouped-Query Attention (GQA), the computational backbone remains consistent. At each layer  $l$ , attention mechanisms read from and write to a central residual stream. Crucially for our intervention methodology, even when keys and values are shared across heads, the output of each individual head,  $O^{(l,h)}$ , is computed and projected back into the residual stream independently. This structural uniformity allows our activation patching to target the localized output projections of individual heads, treating them as discrete causal components that can be isolated across both the base and instruction-tuned states.

### 3.2 Tasks and Dataset Construction

We evaluate mathematical and logical reasoning using a large-scale synthetic dataset of deterministic tasks that isolate sequential updating, boolean aggregation, multi-path routing, and latent feature gating. Let a reasoning task be defined by an input prompt  $X$  and a definitive ground-truth target  $Y$ . Standard few-shot prompting allows the model to generate an unconstrained sequence of intermediate reasoning tokens  $Z = (z_1, z_2, \dots, z_k)$  prior to predicting  $Y$ . To ensure robust statistical power, we scale our dataset to 2,000 total instances (500 instances per task type).

The tasks comprising our dataset include the following four structural primitives:

- **Linear Symbolic:** Sequential arithmetic operations (e.g., "Start with  $x$ . Add  $y$ . Subtract  $z$ .").
- **Parity Computation:** Evaluating the truth value of multiple boolean predicates and determining if the total count of valid predicates is even or odd.
- **Multway Branching:** Executing conditional logic paths based on the modulo of an initial computation (e.g., "If  $S \equiv 0 \pmod{3}$ , return  $x + y$ .").

- **Conditional Branching with Latent Gating (CBLG):** Routing a mathematical operation based on a latent boolean property of a raw input (e.g., "Input:  $a, b$ . If  $a$  is even, calculate  $a/2 + b$ . If  $a$  is odd, calculate  $a - b$ .").

For each task instance, we procedurally generate a clean prompt  $X_c$  with ground-truth answer  $Y_c$ , and a corresponding corrupted prompt  $X_{corr}$  (e.g., modifying a starting value to flip a logic gate) which yields a distinct corrupted answer  $Y_{corr}$ . Our generation constraints guarantee strict digit-width boundaries, ensuring perfect token-length alignment between  $X_c$  and  $X_{corr}$ .

These task types serve as a proxy for algorithmic completeness within the transformer architecture and ensure that our faithfulness metric is not biased toward a single cognitive heuristic. If a model's reasoning circuits are proven faithful across all four of these diverse computational primitives, it provides strong causal evidence that the model is actively executing generalizable control-flow operations rather than relying on shallow, task-specific pattern matching. Unlike causal tracing evaluations on unstructured natural language where the variable position of subject tokens limits interventional scope (Meng et al., 2022), our strictly controlled synthetic templates ensure uniform token alignment, providing mathematically sound conditions for resample ablation.

To extend our evaluation beyond isolated primitives, we also employ a mechanistically aligned GSM8K-Proxy dataset. While the original GSM8K dataset (Cobbe et al., 2021) serves as the standard benchmark for multi-step mathematical reasoning, its highly unstructured, free-form nature precludes rigorous resample ablation. Specifically, modifying real GSM8K instances to create counterfactual pairs ( $X_{corr}$ ) frequently alters the token length of both the prompt and the resulting reasoning trace, leading to catastrophic tensor shape mismatches during activation patching. Our GSM8K-Proxy circumvents this limitation by procedurally generating word problems that mirror the linguistic complexity and multi-step arithmetic of the original dataset, while strictly enforcing the token-width constraints and syntactic symmetry required for clean, bidirectional interventions.

### 3.3 Eliciting Natural Reasoning Traces

To quantify the causal load of specific attention heads, we evaluate the model's autonomous rea-

soning capabilities. We utilize standard few-shot (8-shot) prompting to elicit unconstrained Chain-of-Thought (CoT) generation. The model is provided with the clean prompt  $X_c$  and tasked with generating a sequence of intermediate reasoning tokens  $Z_{natural} = (z_1, z_2, \dots, z_k)$  culminating in a final answer prediction.

Evaluating causal load on mechanically broken rationales yields invalid measurements; ablating a circuit on a task the model already fails to comprehend provides no causal signal. Therefore, we semantically parse each generated trace to extract the model’s final predicted answer. We strictly filter the dataset to include only instances where the model’s autonomous generation perfectly matches the ground-truth target  $Y_c$ . By restricting our analysis to naturally successful traces, we guarantee that the baseline accuracy of our evaluation subset is exactly 100%.

### 3.4 Dynamic Circuit Discovery via Activation Patching

Prior to conducting resample ablation, we require a rigorous method to identify the specific attention-routing circuit ( $S$ ) responsible for successfully executing each task. To achieve this without relying on heuristic architectural guesses, we employ exhaustive single-head activation patching on the filtered subset of successful traces.

To ensure our causal measurement occurs at the exact computational moment the model predicts the target, we dynamically append the model’s own generated CoT prefix to the prompt. We then record a clean activation cache ( $A_c$ ) by running a forward pass on this aligned clean prompt. Next, we execute a forward pass on the corrupted prompt to establish a baseline corrupted Logit Difference (the relative probability of the clean target token versus the corrupted target token).

We systematically sweep through the network, intervening on every attention head individually. For each head, we patch its specific output activation ( $z$ ) in the corrupted run with its corresponding activation from the clean cache  $A_c$ . The causal importance of each head is quantified by the magnitude by which patching that single head restores the Logit Difference toward the correct target.

Crucially, rather than selecting an arbitrary, static number of top- $k$  heads, we aggregate the impact scores and apply a Dynamic Thresholding function. We sort the heads by positive causal impact and dynamically select the minimal subset of heads

required to achieve 80% of the network’s total cumulative restorative impact. This dynamic thresholding intrinsically accounts for architectural differences between the Base and Instruct models, ensuring we isolate the precise functional subgraph regardless of how densely or sparsely the reasoning pathway is distributed.

### 3.5 Resample Ablation and Faithfulness

Having isolated a 100% mechanically coherent baseline and defined the dynamic reasoning circuit  $S$  for each task, we execute the causal intervention. Prior approaches measure faithfulness by perturbing the surface form of chain-of-thought reasoning and observing changes in model outputs (Lanham et al., 2023). In contrast, our method operates directly on internal activations, enabling precise causal analysis of the underlying computation rather than its textual explanation.

To evaluate this causal faithfulness without risking the sequence divergence inherent to open-ended generation or the out-of-distribution manifold collapse associated with zero-ablation (Heimersheim and Nanda, 2024), we employ dual-direction resample ablation.

To ensure exact token-alignment across interventions, we utilize a static prefix alignment strategy. For each successful instance, we append the model’s autonomously generated CoT prefix to both the clean ( $X_c$ ) and corrupted ( $X_{corr}$ ) prompts. We cache the internal activations of both static passes, yielding  $A_c$  and  $A_{corr}$ .

Rather than relying on binary accuracy metrics, which can mask subtle internal shifts, we quantify the causal load of the targeted circuit by measuring the Logit Difference of the target prediction token under two continuous intervention states:

- **Noising:** We perform a forward pass on the clean prompt but actively overwrite the targeted reasoning circuit  $S$  with activations from  $A_{corr}$ :

$$\tilde{A}^{(l,h)} = \begin{cases} A_{corr}^{(l,h)} & \text{if } (l, h) \in S \\ A_c^{(l,h)} & \text{otherwise} \end{cases}$$

A significant drop in the clean Logit Difference indicates the circuit is causally *necessary*.

- **Denoising:** We perform a forward pass on the corrupted prompt but overwrite circuit  $S$  with activations from  $A_c$ . A significant recovery

toward the clean target indicates the circuit is causally *sufficient*.

The joint use of noising and denoising allows us to cross-validate causal claims, identifying components that are both necessary and sufficient for the behavior. Crucially, this bidirectional approach helps account for the inherent asymmetries and potential geometric artifacts commonly introduced by activation patching (Makelov et al., 2024).

## 4 Experiments and Results

### 4.1 Datasets, Models, and Evaluation Setup

We evaluate our pipeline across two distinct dataset regimes to capture both isolated algorithmic routing and natural language generalization. First, we utilize 2,000 instances of our four synthetic task types: Linear Symbolic, Parity Computation, Multiway Branching, and Conditional Branching with Latent Gating (CBLG). Second, to validate our findings in an open-ended natural language context, we evaluate 500 instances of a Mechanistically Aligned GSM8K-Proxy dataset.

To isolate the impact of alignment tuning on internal representations, we conduct a strict within-family comparison using the 1B parameter class of the Llama architecture, specifically comparing the foundation model (Llama-3.2-1B) against its instruction-tuned counterpart (Llama-3.2-1B-Instruct). Because both models share an identical computational backbone and Grouped-Query Attention (GQA) topology, structural deviations in circuit faithfulness can be directly attributed to post-training alignment interventions.

The evaluation proceeds in three phases:

1. **Natural Baseline Generation:** Establishing the model’s autonomous ability to accurately solve the task using unconstrained, 8-shot Chain-of-Thought, yielding our 100% baseline subset.
2. **Dynamic Circuit Discovery:** Identifying the minimal subgraph of attention heads required to account for 80% of the network’s restorative causal impact during single-head activation patching.
3. **Dual-Direction Resample Ablation:** Quantifying the causal load of the discovered circuits via Logit Differences, yielding continuous scores for both Necessity (performance

drop under Noising) and Sufficiency (performance recovery under Denoising).

### 4.2 Circuit Discovery and Topological Sparsity

Before executing causal interventions, our Dynamic Circuit Discovery phase revealed topological differences in how reasoning is distributed across the two models. While both models utilize a dedicated subset of attention heads to route mathematical logic, the instruction-tuned model exhibited a higher degree of circuit diffusion on more complex logical tasks.

For instance, to reach the 80% cumulative causal impact threshold on the Parity task, the Base model relied on a highly modular, sparse subgraph of 68 attention heads. In contrast, the Instruct model required a significantly larger network of 181 heads to reach the exact same restorative threshold. This trend held broadly across the multi-step synthetic suite, with the Instruct model requiring an average of 95 heads per task compared to the Base model’s 57.

Interestingly, on the natural language GSM8K-Proxy, both architectures converged on highly similar head counts (38 and 37 heads, respectively), suggesting that severe circuit diffusion is most pronounced when the aligned model is forced to execute pure, abstract logical routing without the anchoring of natural language word problems. Ultimately, this structural diffusion across the residual stream necessitated the use of dynamically sized intervention hooks for our subsequent resample ablation.

### 4.3 Dual-Direction resample ablation

Applying dual-direction resample ablation to these dynamically extracted circuits exposed distinct causal profiles between the foundation and instruction-tuned architectures. Table 1 details the Necessity and Sufficiency scores across all task distributions.

Across both models and all tasks, the discovered circuits demonstrated near-perfect Sufficiency. When the isolated reasoning subgraphs were injected into corrupted contexts, the Llama Base model recovered 100.36% of its baseline logit difference on synthetic tasks and 98.88% on the GSM8K proxy. Similarly, the Instruct model achieved 100.10% and 98.69% recovery, respectively. This confirms that the dynamically thresholded circuits independently possess the full com-

putational capacity required to execute the target mathematical primitives. However, Noising interventions (corrupting the primary reasoning circuits) revealed a stark divergence in Necessity scores.

For the Llama Base model, completely corrupting the targeted reasoning circuits resulted in minimal performance degradation, yielding a Necessity drop of only 2.92% on synthetic primitives and 4.96% on GSM8K. Conversely, the Instruct model proved significantly more sensitive to internal perturbations. While performing identically on the GSM8K proxy (5.10% drop), the Instruct model exhibited a Necessity drop of 6.79% on the synthetic primitives, which is more than double the degradation observed in the Base model under identical noising conditions.

## 5 Discussion

Our dual-direction resample ablation reveals a topological divergence in how mathematical resilience is distributed between foundation and aligned architectures. In the Llama Base model, completely corrupting the dynamically isolated reasoning circuit resulted in a minimal Necessity drop of only 2.92% on synthetic primitives. Given that this exact same circuit demonstrated near-perfect Sufficiency when evaluated in isolation, this minimal performance degradation initially appears paradoxical.

However, this resilient behavioral plateau perfectly quantifies the Hydra Effect (McGrath et al., 2023) in the context of algorithmic reasoning. The foundation model possesses a highly parallel, distributed mathematical capacity. When early-layer attention heads are corrupted via resample noising, dormant "backup circuits" in later layers dynamically detect the aberrant residual stream and re-compute the missing logical vectors to salvage the generation. Recent feature-level interventions using sparse autoencoders (Chen et al., 2025) have similarly highlighted that this redundant nature of mathematical Chain-of-Thought frequently obscures causal attribution. Our findings confirm that in foundation models, reasoning is not a fragile, linear pathway, but a highly redundant structural web capable of profound self-repair.

Furthermore, our comparison of the Llama-3.2-1B models demonstrates that model self-repair is not a static architectural guarantee and illustrates how alignment tuning can heavily degrade this capability. When subjected to identical noising interventions, the Instruct model exhibited a Necessity

drop of 6.79% on the synthetic primitives—more than double the degradation observed in the Base model.

We posit that this increased structural brittleness represents a mechanistic "Alignment Tax." During Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF), the network is heavily penalized for raw, textbook-style outputs and rewarded for conversational formatting, tone, and instruction compliance. To satisfy these competing multi-objective constraints without increasing the parameter count, the network is forced to repurpose its attention heads. By reallocating its dormant, redundant capacity toward linguistic and formatting adherence, the aligned model centralizes its mathematical computation. Having sacrificed its distributed backup circuits to the Alignment Tax, the Instruct model's primary reasoning pathway loses its self-repair capabilities, rendering it uniquely vulnerable to internal perturbations.

It is important to contextualize the magnitude of this tax. While an absolute Necessity drop of 6.79% is relatively modest and may not immediately signal catastrophic task failure in real-world deployment, the relative shift—more than doubling the architectural brittleness—serves as a vital mechanistic signal. Furthermore, while this structural degradation intersects with broader debates on CoT faithfulness, our measurements strictly isolate the causal necessity and robustness of internal routing, which serves as the physical prerequisite for faithful verbalization.

This hypothesis is further supported by the models' converging behavior on the GSM8K proxy dataset. While the Instruct model was significantly more brittle on the abstract synthetic tasks, both models exhibited nearly identical Necessity drops on the GSM8K proxy (4.96% for Base, 5.10% for Instruct).

Unlike the isolated synthetic primitives, GSM8K word problems require heavy, multi-step linguistic parsing alongside multi-variable mathematical tracking. We hypothesize that these complex, natural language-grounded tasks push the 1B parameter architecture to its representational capacity ceiling. When evaluating the GSM8K proxy, even the Base model must exhaust its available attention heads to simply route the logic, leaving few dormant heads available to act as a self-repairing "Hydra." Consequently, when task complexity scales to exhaust network capacity, foundation models exhibit the

Task Class	Llama-3.2-1B (Base)		Llama-3.2-1B-Instruct	
	Necessity ( $\downarrow$ )	Sufficiency ( $\uparrow$ )	Necessity ( $\downarrow$ )	Sufficiency ( $\uparrow$ )
<i>Mechanistically Aligned Synthetic Primitives</i>				
Linear Symbolic	4.29%	99.73%	8.05%	99.37%
Latent Gating (CBLG)	4.13%	100.65%	7.89%	100.59%
Parity Computation	0.47%	100.62%	0.38%	100.54%
Multiway Branching	0.73%	100.32%	0.78%	100.34%
<b>Synthetic Average</b>	<b>2.92%</b>	<b>100.36%</b>	<b>6.79%</b>	<b>100.10%</b>
<i>Natural Language Proxy</i>				
GSM8K Proxy	4.96%	98.88%	5.10%	98.69%

Table 1: Dual-Direction Activation Patching Logit Differences. Necessity indicates the relative performance drop under noising interventions (corrupting the targeted reasoning circuit), while Sufficiency indicates performance recovery under denoising interventions. The Instruct model exhibits a necessity drop more than double that of the Base model on synthetic primitives, quantifying the degradation of its structural redundancy.

same structural brittleness as their alignment-taxed counterparts.

## 6 Limitations

**Scale and Capacity Starvation:** While our dual-direction resample ablation provides empirical evidence of an alignment-induced brittleness, our experiments are localized to a strict within-family comparison at the 1B parameter scale. At this size, the representational capacity budget may be easily exhausted. It remains an open question whether the "Alignment Tax" is a universal consequence of alignment, or simply an artifact of capacity starvation in smaller architectures. Massive frontier models (e.g., 70B+ parameters) may possess sufficient capacity to absorb multi-objective constraints without sacrificing their distributed backup circuits.

**Intervention Granularity:** To isolate the specific dynamics of logical routing while maintaining precise intervention boundaries, our resample ablation was restricted to attention head outputs ( $z$ ). The residual stream sequentially updates via  $x^{(l+1)} = x^{(l)} + \text{Attn}^{(l)} + \text{MLP}^{(l)}$ . While recent literature highlights the role of Multi-Layer Perceptrons (MLPs) in latent reasoning (Geva et al., 2023), quantifying how alignment independently alters MLP-based representation represents a distinct methodological challenge outside the scope of our attention-focused interventions.

**Monolithic Alignment:** Our methodology compares a foundational base model directly against its final instruction-tuned counterpart. Modern alignment pipelines consist of multiple distinct phases (e.g., Supervised Fine-Tuning followed by RLHF or DPO). Our current experimental design treats

instruction-tuning as monolithic and cannot isolate which specific stage of training is responsible for repurposing the backup circuits.

**Dataset Variance:** To secure the mathematically airtight token-alignment required for causal scrubbing, our evaluation heavily leverages procedurally generated primitives. While these tasks isolate control-flow operations, they lack the linguistic entropy of free-form queries (Dziri et al., 2023). Additionally, we note that performance degradation was not uniformly distributed across the synthetic suite, suggesting that the brittleness of the aligned model may manifest differently depending on the specific cognitive heuristic being evaluated.

**Prompt Limitations:** Our dynamic circuit discovery is inherently conditioned on the specific 8-shot prompt template used to elicit the unconstrained baseline. It remains an open question whether these discovered subgraphs represent universal mathematical circuits, or whether they are partially overfit to the stylistic syntax of the provided prompt, a known challenge in automated circuit discovery (Conmy et al., 2023).

## 7 Conclusion

In this work, we investigated the structural impact of alignment tuning on LLM algorithmic reasoning. Using naturally elicited Chain-of-Thought traces and dual-direction resample ablation on the Llama-3.2-1B architecture, we revealed a notable mechanistic divergence. While foundation models leverage highly redundant, self-repairing computational networks (the Hydra Effect), instruction-tuned models exhibit significantly reduced redundancy. We frame this as the *Alignment Tax on*

*Redundancy*: managing simultaneous logical and linguistic constraints repurposes dormant backup circuits, centralizing reasoning pathways and increasing their sensitivity to internal perturbation.

Although the absolute magnitude of this measured brittleness is modest at the 1B scale, these findings indicate that evaluating model reliability solely on generated text may be insufficient for high-stakes environments, as alignment optimization can mask underlying mathematical brittleness. Furthermore, while precise circuit mapping carries dual-use risks for adversarial weight-editing, it is essential for transparent evaluation. Future work should scale this methodology to frontier architectures—to disentangle this alignment tax from small-model capacity starvation—and isolate the specific impacts of Supervised Fine-Tuning versus preference optimization, ultimately guiding the development of redundancy-preserving alignment techniques that ensure models remain both human-aligned and mechanistically robust.

## References

- Chirag Agarwal, Sree Harsha Tanneru, and Himabindu Lakkaraju. 2024. [Faithfulness vs. plausibility: On the \(un\)reliability of explanations from large language models](#). *arXiv*.
- Pepa Atanasova, Jakob Grue Simonsen, Maria Liakata, and Isabelle Augenstein. 2023. [Faithfulness tests for natural language explanations](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.
- Xi Chen, Aske Plaat, and Niki van Stein. 2025. [How does Chain of Thought think? mechanistic interpretability of Chain-of-Thought reasoning with sparse autoencoding](#). *arXiv*.
- 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*.
- Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. [Towards automated circuit discovery for mechanistic interpretability](#). In *Advances in Neural Information Processing Systems*.
- Nouha Dziri, Zhou Yu, Siva Reddy, and Danqi Chen. 2023. [Faith and fate: Limits of transformers on compositionality](#). In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*.
- 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 (EMNLP)*. Association for Computational Linguistics.
- Stefan Heimersheim and Neel Nanda. 2024. [How to use and interpret activation patching](#). *arXiv preprint arXiv:2404.15255*.
- Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, and 1 others. 2023. [Measuring faithfulness in Chain-of-Thought reasoning](#). In *Advances in Neural Information Processing Systems*.
- Haolang Lu, Hongrui Peng, Weiye Fu, Guoshun Nan, Xinye Cao, Xingrui Li, Hongcan Guo, and Kun Wang. 2026. [Disentangling deception and hallucination failures in LLMs](#). *arXiv*.
- Aleksandar Makelov, Georg Lange, and Neel Nanda. 2024. [Is this the subspace you are looking for? an interpretability illusion for subspace activation patching](#). In *Proceedings of the Twelfth International Conference on Learning Representations*.
- Thomas McGrath, Jacob Kaplansky, Neel Nanda, and 1 others. 2023. [The hydra effect: Emergent self-repair in language model computations](#). *arXiv preprint arXiv:2307.15771*.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. [Locating and editing factual associations in GPT](#). In *Advances in Neural Information Processing Systems*.
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, and 1 others. 2022. [In-context learning and induction heads](#). *Transformer Circuits Thread*.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. [Language models don't always say what they think: Unfaithful explanations in Chain-of-Thought prompting](#). In *Advances in Neural Information Processing Systems*.
- Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. [Interpretability in the wild: A circuit for indirect object identification in GPT-2 small](#). In *International Conference on Learning Representations*.
- Yanbo Wang, Yongcan Yu, Jian Liang, and Ran He. 2025. [A comprehensive survey on trustworthiness in reasoning with large language models](#). *arXiv*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. [Chain-of-Thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*.

- Xi Ye and Greg Durrett. 2022. [The unreliability of explanations in few-shot prompting for textual reasoning](#). In *Advances in Neural Information Processing Systems*.
- Wei Jie Yeo, Ranjan Satapathy, and Erik Cambria. 2025. [Towards faithful natural language explanations: A study using activation patching in large language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10425–10447.
- Kerem Zaman and Shashank Srivastava. 2025. [Is Chain-of-Thought really not explainability? Chain-of-Thought can be faithful without hint verbalization](#). *arXiv*.
- Chunting Zhou, Pengfei Liu, Puxin Xu, and 1 others. 2023. [LIMA: Less is more for alignment](#). In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*.

# bLLeQA: Benchmarking LLMs for Grounded Legal Question-Answering in French and Dutch

Nikolay Banar\*, Ehsan Lotfi\*, Jens Van Nooten,  
Marija Kliocaitė, Walter Daelemans

University of Antwerp, Belgium

Correspondence: nicolae.banari@uantwerpen.be

## Abstract

Retrieval-augmented generation (RAG) systems can play an important role in making law more accessible. However, large and reliable resources for training and benchmarking such systems remain scarce, especially for under-resourced languages like Dutch. To address this gap, and building on previous work (Louis et al., 2024), we introduce bLLeQA, a bilingual parallel question-answering dataset grounded in Belgian legal resources, both in French and Dutch. The dataset contains aligned questions, answers, and supporting articles in both languages, enabling evaluation of both retrieval and end-to-end RAG pipelines. Using bLLeQA, we benchmark the full RAG pipeline in a zero-shot setting, covering retrieval, citation extraction, refusal behavior, and generation quality. Our experiments show that open-weight models are competitive with proprietary models in retrieval and citation extraction, but lag behind in generation quality in the RAG pipeline. Across all models, refusal capability remains weak, meaning that models do not reliably detect when the provided supporting sources are incomplete. In addition, the end-to-end RAG setup still yields a substantial share of flawed responses, reaching 20% even in the best-case scenario.

## 1 Introduction

Access to justice remains a critical challenge for individuals and communities around the world. Disagreements with landlords, workplace disputes, and other civil justice problems are common occurrences, yet a significant portion of the population lacks the knowledge to resolve these issues or understand their rights (Balmer et al., 2010). The primary barriers to accessing justice are the prohibitive costs of legal counsel and a widespread lack of awareness about available legal options (Redelaar et al., 2024). Automated legal question

answering (LQA) systems represent a promising avenue to democratize access to legal information by providing affordable, scalable assistance to broad audiences (Redelaar et al., 2024). LQA involves responding to queries, a task traditionally performed by domain experts, by reviewing relevant laws, interpreting statutes, and applying legal principles to specific facts (Ariai et al., 2025).

The rapid evolution of large language models (LLMs) has significantly advanced the capabilities of LQA systems. These models demonstrate remarkable proficiency in processing large volumes of text and generating human-like responses, offering opportunities to streamline legal research for professionals and lower barriers to information for the general public (Akarajadwong et al., 2025; He et al., 2026). However, the application of LLMs in the legal domain is not without challenges. Legal queries require high precision, and general-purpose LLMs are prone to hallucinations and relying on outdated information. To mitigate this, retrieval-augmented generation (RAG) has become a standard framework for reliable LQA. By retrieving authoritative legal sources to ground the model’s generation, RAG systems ensure that answers are verifiable, auditable, and based on up-to-date legal texts (He et al., 2026).

Despite the proliferation of LQA systems, there is a significant disparity in their availability across different languages and jurisdictions. Most existing resources and benchmarks focus on resource-rich languages such as English and common law traditions. However, legal systems are inherently jurisdiction-specific; a digital legal aid system designed for the US is ineffective for a civil law country operating under a different framework and language, and there is a critical need for LQA solutions that can handle the nuances of local law (Redelaar et al., 2024).

As a multilingual country, Belgium invests significant resources to consolidate its laws in both

\*indicates equal contribution

French and Dutch, using qualified legal professionals. This results in a highly valuable resource for research in multilingual legal applications. Leveraging this potential and building on the Belgian LLeQA dataset (Louis et al., 2024) in French, we introduce the bilingual LLeQA (bLLeQA) that extends LLeQA to a parallel French-Dutch setting, via alignment, translation, and refinement of the original annotations in collaboration with a legal professional. Using bLLeQA, we conducted extensive benchmarking of LLMs in three RAG scenarios, providing insights into the performance of different LLMs in handling legal queries and citations. Our contributions are the following:

- We create and publish a parallel bilingual dataset for retrieval-based legal question-answering in French and Dutch, based on Belgian legislation.<sup>1</sup>
- We benchmark a wide range of open and proprietary LLMs on legal question-answering in French and Dutch, under different RAG settings.<sup>2</sup>

## 2 Related Work

Existing resources for legal question answering can be categorized by the scope and complexity of the task, ranging from document retrieval to answer generation and full RAG evaluation.

**Legal Retrieval** Legal document retrieval concerns the task of finding and ranking documents relevant to a given query from a large set of candidates. It is commonly studied over a corpus of cases (legal case retrieval) or articles (statutory article retrieval). In case retrieval, LeCaRD (Ma et al., 2021) (later extended as LeCaRDv2 (Li et al., 2024)) provided one of the first resources for the Chinese legal system, while ECtHR-PCR (T.y.s.s. et al., 2024) and CLERC (Hou et al., 2025) offered similar datasets for European and US jurisdictions, respectively. In statutory article retrieval, examples include BSARD (Louis and Spanakis, 2022), bBSARD (Lotfi et al., 2025b), and STARD (Su et al., 2024), which provided sizable resources based on Belgian and Chinese legislation in French, Dutch, and Chinese. In these datasets, each legal query is labeled with a set of articles that human annotators identified as necessary to answer the query.

**LQA** LQA is the task of providing valid responses to legal queries, with or without having access to legal documents (open- or closed-book). The response can be extracted from a given context (CJRC (Duan et al., 2019), EQUALS (Chen et al., 2023)), selected from provided options or choices (JEC-QA (Zhong et al., 2020), PIL-QA (Sovrano et al., 2021), JuRO (Craciun et al., 2025)), retrieved from a response bank (LegalQA (Askari et al., 2024), FALQU (Mansouri and Campos, 2023)) or generated from scratch. Examples of the latter include cLegal-QA (Wang et al., 2024c) and LeDQA (Liu et al., 2024) for Chinese, PrivacyQA (Ravichander et al., 2019) for English, LEGAL-UQA (Faisal and Yousaf, 2024) for Urdu, and GerLayQA (Büttner and Habernal, 2024) for German. These datasets often contribute to large legal benchmarks like LegalBench (Guha et al., 2023), which attempt to assess the legal knowledge and reasoning abilities of LLMs in a closed-book setting.

**Legal RAG** With the rapid rise of LLMs, RAG has emerged as a crucial method for improving the factual accuracy and interpretability of LQA systems (He et al., 2026). Combining the retrieval and generation steps, RAG aims to ground the response of an LLM in the retrieved context, making it a specific instance of open-book question answering (QA). Existing RAG resources usually consist of QA pairs (synthetic or human-written) labeled with relevant articles or passages from a large corpus of legal documents. Examples include LLeQA (Louis et al., 2024) for French, NitiBench (Akarajaradwong et al., 2025) for Vietnamese, and ObliQA (Gokhan et al., 2025) together with its multi-passage version ObliQA-MP (Gokhan and Briscoe, 2025) for English. For benchmarking, LegalBench-RAG (Pipitone and Alami, 2024) was proposed to assess precise retrieval by focusing on extracting minimal, highly relevant text segments from legal documents.

Our work builds on LLeQA (Louis et al., 2024), a long-form legal question-answering dataset in French that was primarily created by adding human-written responses to the BSARD (Louis and Spanakis, 2022) retrieval dataset. Similarly to bBSARD (Lotfi et al., 2025b) which extends BSARD to a bilingual French-Dutch setting, in this work we align, translate, and refine LLeQA annotations to create bLLeQA, and then use it to benchmark LLMs on LQA.

<sup>1</sup><https://huggingface.co/datasets/clips/bLLeQA>

<sup>2</sup><https://github.com/nikolay-banar/blleqa>

### 3 Dataset

In this section, we describe how bLLeQA was constructed from LLeQA (Louis et al., 2024). LLeQA comprises 1,868 expert-annotated legal questions in the French language, along with answers grounded in Belgian legislation (~28k articles). The dataset was curated in collaboration with Droits Quotidiens<sup>3</sup>, a Belgian non-profit organization that aims to make the law comprehensible and accessible to the public, and to this end, maintains a rich website featuring legal questions commonly posed by Belgian citizens. Each question comes with one or more categories, references to relevant legislative statutes, and a detailed answer written in layman’s terms by experienced legal experts.

To create a parallel bilingual dataset from LLeQA, we follow these steps: (i) we extract and align the Dutch version of legal codes to build a parallel corpus; (ii) we leverage a combination of automatic translation and expert post-editing to translate questions and answers into Dutch; (iii) we ask a bilingual legal expert to ensure that answers are indeed grounded in the provided articles. Since LLeQA is based on BSARD, which has already been extended to a bilingual version (bBSARD (Lotfi et al., 2025b)), we take advantage of this resource where possible. These steps are described in more detail below.

**Corpus Alignment** The parallel bBSARD retrieval corpus covers 79% of the articles in LLeQA. For the rest, we scraped approximately 6,000 French-Dutch article pairs from the Justel Database<sup>4</sup> to ensure that both language versions correspond to the same official legal provisions. These steps resulted in an alignment rate of 93% (25,982 out of 27,942) articles in both languages. None of the missing articles are cited in the answers.

**QA Pairs** Similarly to the previous step, we started with bBSARD, which covers 40% of the questions in LLeQA. Translations for the remaining questions and all reference answers were generated using GPT-5.0. Then a bilingual legal expert (native French and Dutch speaker) was asked to review the translations and rectify any potential issues<sup>5</sup>.

<sup>3</sup><https://www.droitsquotidiens.be/fr>

<sup>4</sup><https://www.ejustice.just.fgov.be>

<sup>5</sup>In total, 20.8% of the reference answers and 1.4% of the questions needed corrections. Most common issues included failing to translate abbreviations (e.g. 'MENA') and to identify Flemish equivalents for Walloon institutions (e.g.

Annotations	Train	Val	Test	All
Initial dataset (#)	1,472	201	195	1868
No changes	70.3%	80.1%	70.3%	72.0%
Corrected	6.2%	10.0%	9.2%	6.3%
Removed	23.5%	9.9%	20.5%	21.7%
subject mismatch	2.1%	3.5%	8.7%	2.9%
general context	0.1%	0%	5.1%	0.6%
missing information	0.5%	0.5%	3.6%	0.8%
legal type mismatch	0.7%	0.5%	3.1%	0.9%
very long context	19.8%	5.5%	0%	16.2%
Final dataset (#)	1,125	181	155	1461

Table 1: RAG annotation outcomes across dataset splits.

**RAG Annotations** For the RAG setup, we ask the legal expert to check whether the cited articles provide sufficient context to produce an answer for each query. As Table 1 shows, 72% of the samples pass this step unchanged (accurate grounding). Over-citing samples (6.3%) were corrected by removing the unnecessary articles, and samples with more serious issues or an excessively long context were removed (21.7%). Most common issues include: (i) *subject mismatch*, where the cited articles do not address the legal subject of the question; (ii) *overly general context*, where the cited articles lack the specificity required to support the answer; (iii) *missing information*, where key information required to justify the answer is absent from the cited articles; (iv) *legal type mismatch*, where the cited articles concern procedural law while the question targets substantive provisions, or vice versa.

The final dataset comprises 1,461 QA pairs (for each language) grounded in a corpus of 25,982 articles of Belgian legislation. Appendix A contains additional details about the dataset.

## 4 Experimental Setup

This section describes the experimental setup for benchmarking retrieval, reranking, and end-to-end RAG on bLLeQA. All experiments are conducted in a zero-shot setting on the test set.

### 4.1 Retrieval and Reranking

**Retrieval** Retrieval experiments are based on the code<sup>6</sup> from bBSARD (Lotfi et al., 2025b). We select a wide range of models, from lexical approaches to static embeddings (i.e., word2vec and fastText), and zero-shot dense retrievers. The complete list of models and the prompts used for the instruct models are provided in Tables 13 and

<sup>6</sup>FAMIWAL’).

<sup>6</sup><https://github.com/nerses28/bBSARD>

15 (Appendix D and E), respectively. For models with a maximum input length of 512 tokens, texts are split into overlapping chunks of 200 tokens with a 20-token overlap. Embeddings from each chunk are aggregated using mean pooling (except for LaBSE which uses the [CLS] token). Then, cosine similarity is computed to score the resulting embeddings.

**Reranking** In addition, we benchmark a number of reranking models (see Table 13 in Appendix D) using the top 100 articles retrieved by BM25, the E5 suite, and Voyage.

**Evaluation Metrics** The models were evaluated using conventional retrieval metrics: macro-averaged recall@k (R@k), mean average precision@k (MAP@k), mean reciprocal rank@k (MRR@k), and normalized discounted cumulative gain@k (nDCG@k).

## 4.2 Question Answering

**Generating Answers** We prompt a wide selection of LLMs, both open-weight and proprietary (see Table 13 in Appendix D), to generate answers to Dutch and French legal questions. For each question, we experiment with three different context settings:

1. *Gold*: A setting where we provide only the *gold-standard context*. We use this setting as a baseline to validate whether the models can follow the instructions and use correct articles.
2. *RAG*: We conduct retrieval on the Dutch corpus with voyage-3-large and use the retrieved *top 100* article IDs for both French and Dutch. The context contains all relevant articles in 71% of cases.
3. *RAG+*: Since the *top 100* retrieved articles do not always include the full gold context ( $\text{Recall}@100 < 1$ ), we add any missing gold-standard articles to the context by replacing an equal number of randomly selected retrieved articles.

The prompt for the models, which remains consistent across settings and models, can be found in Table 16, Appendix E. We do not apply any truncation policy, as the full context in all settings fits within the context window of the tested LLMs. Importantly, we instruct models to (i) refuse to answer a question when the context is insufficient and (ii) answer in paragraphs supported by one or more articles from the provided context. We use

the OpenRouter API<sup>7</sup> to query the models.

**Evaluation Metrics** We evaluate the output of the models on four different aspects.

*Correctness*: We leverage DeepEval’s G-Eval metric (Yang et al., 2024; Liu et al., 2023) to evaluate answer correctness, using an LLM-as-judge with a custom prompt (cf. Table 14, Appendix E). In essence, the LLM is tasked with assigning a score to model outputs ranging from 1 to 5 (1: Critical Failure/Incorrect, 2: Poor/Significant Omissions, 3: Acceptable/Partially Complete, 4: Good/Mostly Accurate, 5: Excellent/Semantically Equivalent), given the gold standard answer. We report averages across all queries per model.

*Faithfulness*: We evaluate the outputs using RAGAS’ faithfulness metric (Es et al., 2024). This metric leverages an LLM-as-judge to extract statements from a model’s answer, and verify the proportion of statements that are supported by the context to produce a score in  $[0, 1]$ . We compute the metric only with respect to the articles cited by the tested models.

*Citation*: Since models are prompted to explicitly ground their statements in context articles, we can compare the cited articles with the gold set to calculate precision, recall, and F1 scores. These metrics provide insight into how well models are able to extract the relevant information from the provided context.

*Refusal*: We also prompt the models to answer the question only if the context is sufficient, and refuse otherwise. This can be used to assess the models’ ability in determining whether the provided context is adequate to answer a question, which is a desirable feature, especially when dealing with sensitive and potentially consequential domains like law. Comparing the true incomplete (or inadequate) contexts with the refused responses, we calculate and report *Precision*, *Recall*, and *F1* scores.

In all settings, correct and incorrect refusal are automatically scored with the upper and lower bound of the correctness and citation scales, respectively: a correct refusal receives the maximum scores for citation coverage (1) and correctness (5), whereas an incorrect refusal receives the minimum scores of 0 for citation coverage and 1 for correctness.

<sup>7</sup><https://openrouter.ai/>

Model	Size	French			Dutch		
		R@100	MAP@100	NDCG@100	R@100	MAP@100	NDCG@100
BM25	-	57.04	19.25	28.44	48.52	15.46	22.93
BM25 + BGE-reranker-v2-m3	-	57.04	27.01	35.24	48.52	22.92	29.93
mE5-small	118M	58.44	19.09	28.10	60.64	21.87	30.85
mE5-base	278M	58.94	20.71	29.77	61.53	21.03	30.48
mE5-large	560M	66.84	27.80	37.31	68.01	25.64	35.94
mE5-large-instruct	560M	73.46	22.49	34.30	71.37	26.02	36.97
E5-mistral-7b	7B	74.30	<b>36.91</b>	46.45	74.73	<b>36.37</b>	<b>46.27</b>
BGE-Mult.-Gemma2	9B	79.86	36.22	<b>48.01</b>	79.59	31.25	43.57
voyage-3-large	-	<b>81.84</b>	34.16	46.51	76.55	32.46	43.65
embedding-3-large	-	79.05	33.61	45.33	<b>80.94</b>	27.54	40.66

Table 2: Retrieval performance of selected models.

**Selecting a Judge Model** Choosing an appropriate judge model is quintessential for accurately estimating a model’s RAG capabilities. To ensure that the model’s assessments align with human judgment, we use all responses produced by DeepSeek-v3.2 in the *Gold* setting<sup>8</sup>, and asked a legal expert to score them from 1 to 5, following the instructions in the answer correctness prompt (Table 14, Appendix E). We then prompt a list of candidate LLMs<sup>9</sup> in the same way, and calculate Spearman correlation, mean average error (MAE) and F1-macro (for the binarized correctness scale), with respect to the expert annotations. In our experiment, Gemini-3-Flash achieves a strong correlation, highest F1-macro, and lowest MAE, for both French and Dutch, and therefore is chosen as the judge. The results for all candidates can be found in Table 5, Appendix B.

## 5 Results and Discussion

In this section, we present and discuss the main experimental results for legal RAG, in both retrieval and generation. In particular, we examine differences in performance between proprietary and open-weight models.

### 5.1 Retrieval

Table 2 shows the retrieval results for the selected models (detailed results are provided in Tables 6 and 7 in Appendix C). BM25 achieves performance comparable to multilingual E5-small in French, but performs worse than it in Dutch. When combined with a reranker, BM25 outperforms multilingual E5-base by a large margin in French and performs comparably in Dutch (MAP@100 and

NDCG@100). In general, model performance tends to improve with model size in both languages. Interestingly, large open-weight models perform comparably to, and in some cases better than proprietary models in both languages.

### 5.2 Generation

In this section, we discuss the generative performance of LLMs in terms of the correctness and faithfulness of the response, citation recall, and refusal, as described in Section 4.2.

**Robustness to Noise** Figure 1 visualizes the quality of answer generation per model and setup. To better understand the results, we divide the responses into 4 categories: correct and incorrect refusals, and accurate and inaccurate answers. To be considered accurate, an answer should score 4 or higher on the 1-5 correctness scale (see the evaluation prompt in Table 14, Appendix E).

Surprisingly, the *Gold* setting does not yield the highest proportion of accurate answers across all models. In many cases, when noise (i.e. irrelevant articles) is introduced into the context in RAG+, the proportion of accurate answers increases (e.g. Qwen3.5-27B, GLM-5, Claude-Sonnet-4.6). This effect might be attributed to lower refusal rates when a longer context is provided. This can suggest a context size bias, where a longer context has a higher chance to pass as sufficient, even if it contains the same necessary information as a significantly shorter context.

In the realistic RAG setting, we observe a slight decrease in the proportion of accurate answers for many models compared with the RAG+ setup (e.g. GLM-5, GPT-5.4). However, for models with relatively strong refusal capabilities, such as GLM-5 and GPT-5.4, the combined proportion of accurate answers and correct refusals can match or exceed the accurate answer rate observed in the earlier setups, where correct refusals do not apply. Hence,

<sup>8</sup>We chose this setting to isolate the answer quality from retrieval errors.

<sup>9</sup>We select recent, high-performing LLMs, as well as models identified in recent studies on LLM-as-a-judge, including Han et al. (2026) and Feng et al. (2025).



Figure 1: Performance of models on the test set under different context settings for Dutch (top) and French (bottom). Table 10 in Appendix C contains the exact results that correspond to this figure.

we can conclude that modern state-of-the-art models of different sizes are generally robust to noise and can perform comparably across settings with larger contexts.

**Citation** Figure 2 shows the average answer correctness score against citation recall under the three context settings, for Dutch and French. As can be observed, there is a strong positive correlation between the two, especially under the RAG and RAG+ settings. While in general this trend adheres to the performance-size relation (i.e. larger models doing better), there are exceptions, most notably the Qwen3.5 family, which performs surprisingly well for its size. In particular, the 27B version achieves relatively high recall scores in a realistic RAG setting, comparable to the largest open models (e.g. GLM-5) and even some state-of-the-art proprietary models (e.g. Gemini-3.1-Pro). This suggests a cost-effective strategy for RAG design by leveraging their citation selection capability, while delegating the generation part to a larger model.

**Generation Quality** Figure 3 plots the average answer correctness score against the corresponding average faithfulness score, under the three context settings, for Dutch and French. Overall, higher faithfulness is associated with higher correctness, but the relationship shows substantial local variation across models and settings. Consequently, better overall performance is indicated by moving toward the upper-right corner (high faithfulness and high correctness).

For French and Dutch, we observe that proprietary models (GPT-5.4, Claude-Sonnet-4.6, Gemini-3-Flash) achieve the highest performance in the RAG setup, combining strong faithfulness and correctness scores. The best observed mean correctness is 4.2, corresponding to the “Good/Mostly Accurate” category. At the same time, roughly 20% of the outputs are flawed, i.e., inaccurate answers or incorrect refusals. The best open-weight models (GLM-5, Qwen3.5-397B-A17B, Kimi-K2-Thinking) achieve faithfulness scores comparable to the strongest proprietary models, but

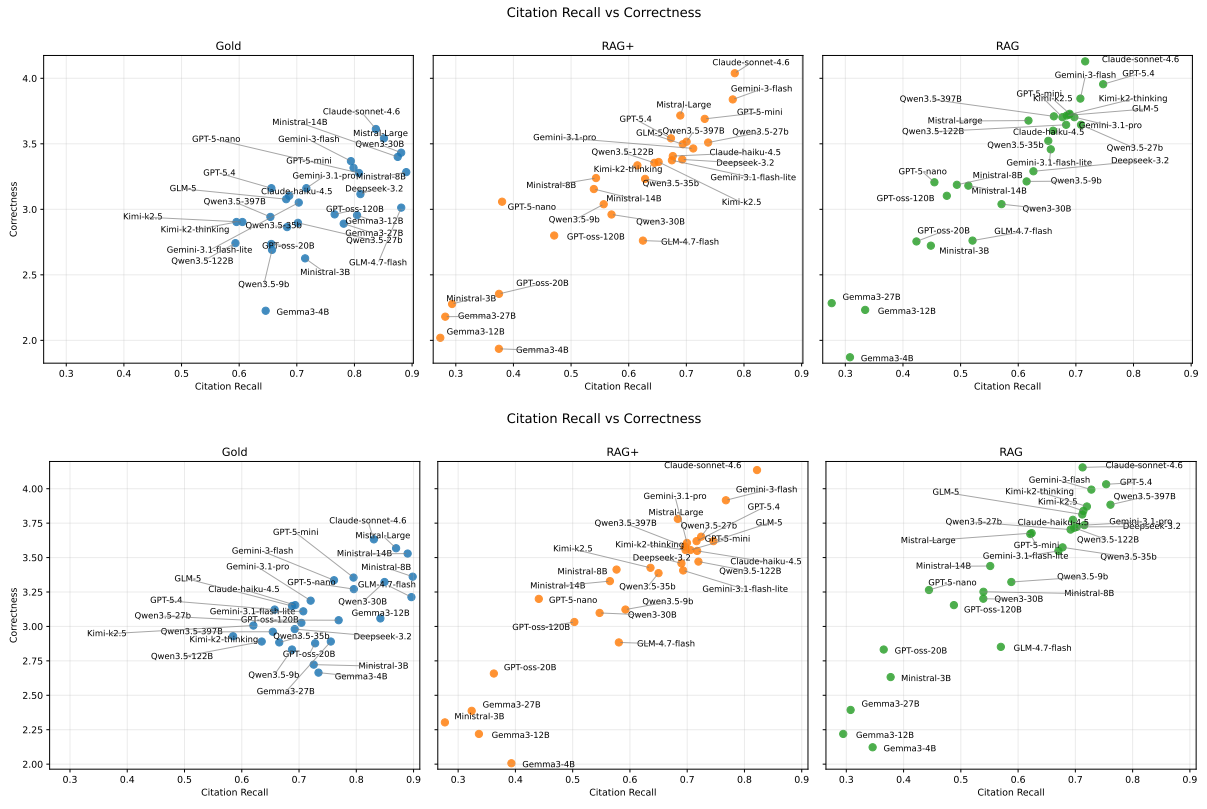


Figure 2: Answer correctness (Table 9, Appendix C) vs. citation recall (Table 8, Appendix C) for different models under the 3 context settings, for Dutch (top) and French (bottom).

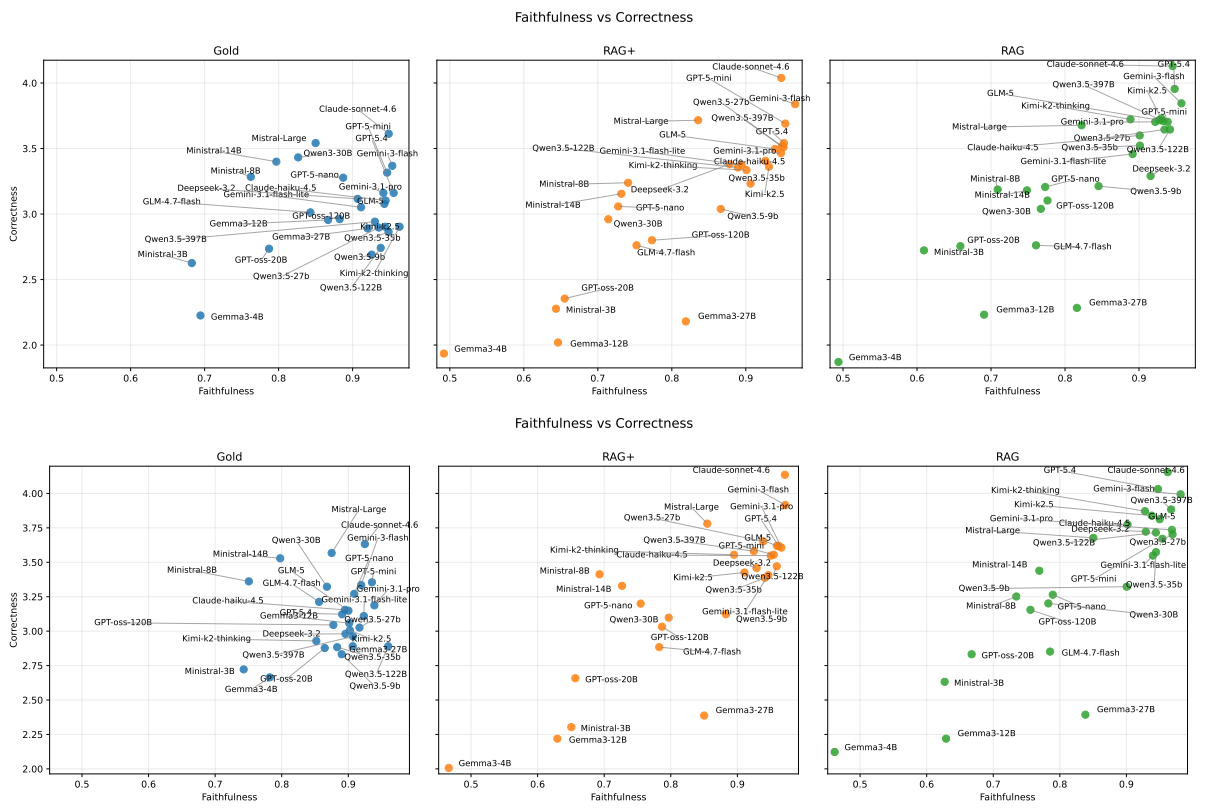


Figure 3: Answer correctness vs. answer faithfulness (Table 9, Appendix C) for different models under the 3 context settings, for Dutch (top) and French (bottom).

this does not translate into the same level of answer quality: their average correctness remains below 4 (the “Good/Mostly Accurate” threshold). Their best-case correctness scores are 3.9 for French and 3.7 for Dutch. For these models, the share of flawed outputs is around 35%. Hence, we observe a large gap between the best proprietary and open-weight models in the real-world setup for both Dutch and French. At the same time, the best open-weight models match or outperform smaller proprietary models such as GPT-5-mini and Claude-Haiku-4.5.

For both languages, we observe strong results for Qwen3.5-family models in all settings. Starting from the 27B version, the Qwen3.5-family combines high faithfulness with strong correctness of 3.7 for French and Dutch, placing the results close to the “Good/Mostly Accurate” category.

**Refusal** We observe that all models struggle to detect incomplete contexts, resulting in poor overall refusal performance (see Tables 11 and 12, Appendix C). The best proprietary (GPT-5.4) and open (Kimi-K2.5) models reach RAG refusal F1 scores of around 56 and 52, respectively. Other models demonstrate even less promising performances. This behavior in refusal handling or context completeness detection is not unique to our case and has also been observed in other tasks (Xu et al., 2025; Sun et al., 2025; Zhou et al., 2026; Kirichenko et al., 2025). Notably, poor refusal performance does not necessarily imply low correctness, as two of the three best-performing models in the RAG setup, Claude-Sonnet-4.6 and Gemini-3-Flash, exhibit low refusal performance. The models’ strong performance in faithfulness indicates that they ground their responses in the provided articles rather than drawing on parametric memory. This suggests that the missing ground-truth articles in the context are not equally important, and in some cases accurate answers can be generated using an incomplete context. Despite this, such behavior can be problematic, as models may overlook important details and nuances that are particularly important in the legal domain.

## 6 Conclusions and Future Work

In this paper, we presented bLLeQA, a parallel bilingual dataset for retrieval-based LQA in French and Dutch. Based on the LLeQA dataset, it comprises 1461 QA pairs (for each language) grounded in a corpus of 25,982 articles of Belgian legisla-

tion, providing a valuable resource for LQA studies in French and Dutch. Using bLLeQA, we evaluated a wide range of open and proprietary LLMs on legal question-answering in both languages, under three different RAG settings, assessing their ability to generate correct and grounded responses, or to refuse when the provided context is insufficient. Open-weight models are competitive with proprietary models in retrieval and citation extraction but still lag behind the strongest proprietary models in response generation. Across all models, correct refusal capability remains weak: models do not reliably detect when the provided sources are incomplete, and this failure leads to performance degradation and an increased proportion of erroneous answers. Moreover, even in the best-case scenario, the end-to-end RAG setup yields a substantial proportion of flawed responses (20%). We believe these findings provide guidance for designing RAG-based LQA systems in French and Dutch.

There are many avenues worth further exploration. Most interesting to us is a detailed error analysis of hallucinations in LLMs: while partly captured by *faithfulness* and *correctness*, it warrants more targeted study for a finer-grained assessment. In addition, we did not explore fine-tuning or agentic setups. Both directions could help address the failure modes observed in our naive RAG pipeline, for example by improving citation extraction and refusal reliability, or by enabling retrieval and verification steps to reduce unsupported generations under noisy or incomplete context.

## Limitations

bLLeQA offers limited coverage of Belgian law, focusing on selected codes from federal and Walloon legislation. In addition, it reflects a specific time slice corresponding to when the original LLeQA dataset was constructed. Given these limitations, bLLeQA is not intended to provide comprehensive legal information or advice. Instead, its primary purpose is to benchmark retrieval and generative models grounded in the provided sources and to gain insights into the current state of the art.

## Acknowledgements

This research received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme. We thank legal professional Manon Quinet for her assistance with data annotation. In addition, we

acknowledge the use of ChatGPT for assisting with error checking and proofreading of this paper.

## References

- Pawitsapak Akarajaradwong, Pirat Pothavorn, Chompakorn Chaksangchaichot, Panuthep Tasawong, Thitiwat Nopparatbundit, Keerakiat Pratai, and Sarana Nutanong. 2025. [NitiBench: Benchmarking LLM frameworks on Thai legal question answering capabilities](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34304–34327, Suzhou, China. Association for Computational Linguistics.
- Anthropic. 2025. Claude haiku 4.5. <https://www.anthropic.com/claude/haiku>. Accessed: 2026-02-09.
- Anthropic. 2026. Introducing claude sonnet 4.6. <https://www.anthropic.com/news/claude-sonnet-4-6>. Accessed: 2026-03-27.
- Farid Ariai, Joel Mackenzie, and Gianluca Demartini. 2025. [Natural language processing for the legal domain: A survey of tasks, datasets, models, and challenges](#). *ACM Comput. Surv.*, 58(6).
- Arian Askari, Zihui Yang, Zhaochun Ren, and Suzan Verberne. 2024. [Answer retrieval in legal community question answering](#). In *Advances in Information Retrieval: 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, 2024, Proceedings, Part III*, page 477–485, Berlin, Heidelberg. Springer-Verlag.
- {Nigel J.} Balmer, Ash Patel, Alexy Buck, Catrina Denvir, and Pascoe Pleasence. 2010. *Knowledge, Capacity and the Experience of Rights Problems*. Public Legal Education Network: PLENet.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.
- Marius Büttner and Ivan Habernal. 2024. [Answering legal questions from laymen in German civil law system](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2015–2027, St. Julian’s, Malta. Association for Computational Linguistics.
- Andong Chen, Feng Yao, Xinyan Zhao, Yating Zhang, Changlong Sun, Yun Liu, and Weixing Shen. 2023. [Equals: A real-world dataset for legal question answering via reading chinese laws](#). In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law, ICAIL ’23*, page 71–80, New York, NY, USA. Association for Computing Machinery.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. [Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). *Preprint*, arXiv:2402.03216.
- Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024b. [M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.
- Cristian-George Craciun, Răzvan-Alexandru Smădu, Dumitru-Clementin Cercel, and Mihaela-Claudia Cercel. 2025. [GRAF: Graph retrieval augmented by facts for Romanian legal multi-choice question answering](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12708–12742, Vienna, Austria. Association for Computational Linguistics.
- DeepMind. 2025a. Gemini 2.5 flash model card. <https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Flash-Lite-Model-Card.pdf>. Accessed: 2026-03-27.
- DeepMind. 2025b. Gemini 3 flash model card. <https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Flash-Model-Card.pdf>. Accessed: 2026-02-10.
- DeepMind. 2025c. Gemini 3 pro model card. <https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf>. Accessed: 2026-02-10.
- DeepMind. 2026. Gemini 3.1 flash-lite model card. <https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-1-Flash-Lite-Model-Card.pdf>. Accessed: 2026-03-27.
- DeepSeek-AI. 2025. Deepseek-v3.2: Pushing the frontier of open large language models.
- Xingyi Duan, Baoxin Wang, Ziyue Wang, Wentao Ma, Yiming Cui, Dayong Wu, Shijin Wang, Ting Liu, Tianxiang Huo, Zhen Hu, Heng Wang, and Zhiyuan Liu. 2019. *CJRC: A Reliable Human-Annotated Benchmark DataSet for Chinese Judicial Reading Comprehension*, page 439–451. Springer International Publishing.
- Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. [RAGAs: Automated evaluation of retrieval augmented generation](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

- Faizan Faisal and Umair Yousaf. 2024. **Legal-uqa: A low-resource urdu-english dataset for legal question answering.** *Preprint*, arXiv:2410.13013.
- Jean-Philippe Fauconnier. 2015. **French word embeddings.** Accessed: 2026-01-05.
- 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.
- Yuanning Feng, Sinan Wang, Zhengxiang Cheng, Yao Wan, and Dongping Chen. 2025. **Are we on the right way to assessing llm-as-a-judge?** *Preprint*, arXiv:2512.16041.
- GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, and 168 others. 2026. **GLM-5: from vibe coding to agentic engineering.** *Preprint*, arXiv:2602.15763.
- Tuba Gokhan and Ted Briscoe. 2025. **Grounded answers from multi-passage regulations: Learning-to-rank for regulatory RAG.** In *Proceedings of the Natural Legal Language Processing Workshop 2025*, pages 135–146, Suzhou, China. Association for Computational Linguistics.
- Tuba Gokhan, Kexin Wang, Iryna Gurevych, and Ted Briscoe. 2025. **Shared task RIRAG-2025: Regulatory information retrieval and answer generation.** In *Proceedings of the 1st Regulatory NLP Workshop (RegNLP 2025)*, pages 1–4, Abu Dhabi, UAE. Association for Computational Linguistics.
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. **Learning word vectors for 157 languages.** In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023. **Legalbench: a collaboratively built benchmark for measuring legal reasoning in large language models.** In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA. Curran Associates Inc.
- Steve Han, Gilberto Titericz Junior, Tom Balough, and Wenfei Zhou. 2026. **Judge’s verdict: A comprehensive analysis of LLM judge capability through human agreement.**
- Congqing He, Haichuan Hu, Yanli Li, Hao Zhang, and Qunjun Zhang. 2026. **A survey of large language models for legal tasks: Progress, prospects and challenges.** *Computer Science Review*, 60:100906.
- Abe Bohan Hou, Orion Weller, Guanghui Qin, Eugene Yang, Dawn Lawrie, Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. 2025. **CLERC: A dataset for U. S. legal case retrieval and retrieval-augmented analysis generation.** In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7898–7913, Albuquerque, New Mexico. Association for Computational Linguistics.
- 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.
- JinaAI. 2025. **jinaai/jina-reranker-v2-base-multilingual.** <https://huggingface.co/jinaai/jina-reranker-v2-base-multilingual>. Accessed: 2026-02-09.
- Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J. Bell. 2025. **Abstentionbench: Reasoning LLMs fail on unanswerable questions.** In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*.
- Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. **Making large language models a better foundation for dense retrieval.** *Preprint*, arXiv:2312.15503.
- Haitao Li, Yunqiu Shao, Yueyue Wu, Qingyao Ai, Yixiao Ma, and Yiqun Liu. 2024. **Lecardv2: A large-scale chinese legal case retrieval dataset.** In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '24*, page 2251–2260, New York, NY, USA. Association for Computing Machinery.
- Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, and 101 others. 2026. **Minstral 3.** *Preprint*, arXiv:2601.08584.
- Bulou Liu, Zhenhao Zhu, Qingyao Ai, Yiqun Liu, and Yueyue Wu. 2024. **Ledqa: A chinese legal case document-based question answering dataset.** In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM '24*, page 5385–5389, New York, NY, USA. Association for Computing Machinery.

- 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.
- Ehsan Lotfi, Nikolay Banar, and Walter Daelemans. 2025a. **BEIR-NL: Zero-shot information retrieval benchmark for the Dutch language**. In *Proceedings of the 18th Workshop on Building and Using Comparable Corpora (BUCC)*, pages 36–45, Abu Dhabi, UAE. Association for Computational Linguistics.
- Ehsan Lotfi, Nikolay Banar, Nerses Yuzbashyan, and Walter Daelemans. 2025b. **Bilingual BSARD: Extending statutory article retrieval to Dutch**. In *Proceedings of the 1st Regulatory NLP Workshop (Reg-NLP 2025)*, pages 10–21, Abu Dhabi, UAE. Association for Computational Linguistics.
- Antoine Louis, Vageesh Kumar Saxena, Gijs van Dijck, and Gerasimos Spanakis. 2025. **ColBERT-XM: A modular multi-vector representation model for zero-shot multilingual information retrieval**. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4370–4383, Abu Dhabi, UAE. Association for Computational Linguistics.
- Antoine Louis and Gerasimos Spanakis. 2022. **A statutory article retrieval dataset in French**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6789–6803, Dublin, Ireland. Association for Computational Linguistics.
- Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2024. **Interpretable long-form legal question answering with retrieval-augmented large language models**. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(20):22266–22275.
- Yixiao Ma, Yunqiu Shao, Yueyue Wu, Yiqun Liu, Ruizhe Zhang, Min Zhang, and Shaoping Ma. 2021. **Lecard: A legal case retrieval dataset for chinese law system**. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21*, page 2342–2348, New York, NY, USA. Association for Computing Machinery.
- Behrooz Mansouri and Ricardo Campos. 2023. **Falqu: Finding answers to legal questions**. *Preprint*, arXiv:2304.05611.
- Meta. 2025. **Llama-3.3-70b-instruct model card**. <https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct>. Accessed: 2026-02-10.
- Meta. 2026. **The llama 4 herd: The beginning of a new era of natively multimodal ai innovation**. <https://ai.meta.com/blog/llama-4-multimodal-intelligence/>. Accessed: 2026-03-27.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. **Efficient estimation of word representations in vector space**. *Preprint*, arXiv:1301.3781.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. **Distributed representations of words and phrases and their compositionality**. *Advances in neural information processing systems*, 26.
- Mistral AI. 2025. **Mistral large 3-675b instruct-2512 model card**. <https://huggingface.co/mistralai/Mistral-Large-3-675B-Instruct-2512>. Accessed: 2026-02-10.
- Moonshot AI. 2026. **Kimi k2.5: Ai that sees, codes, and works like an expert**. <https://www.kimi.com/ai-models/kimi-k2-5>. Accessed: 2026-03-27.
- OpenAI. 2025. **Gpt-5 mini model**. <https://platform.openai.com/docs/models/gpt-5-mini>. Accessed: 2026-02-10.
- OpenAI. 2025a. **gpt-oss-120b & gpt-oss-20b model card**. *Preprint*, arXiv:2508.10925.
- OpenAI. 2025b. **Introducing gpt-5**. <https://openai.com/index/introducing-gpt-5/>. Accessed: 2025-11-13.
- OpenAI. 2025c. **Introducing gpt-5.4**. <https://openai.com/index/us-EN/introducing-gpt-5.4/>. Accessed: 2026-03-27.
- Nicholas Pipitone and Ghita Houir Alami. 2024. **Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain**. *Preprint*, arXiv:2408.10343.
- QwenTeam. 2026. **Qwen3.5: Towards native multimodal agents**. <https://qwen.ai/blog?id=qwen3.5>. Accessed: 2026-03-27.
- Abhilasha Ravichander, Alan W Black, Shomir Wilson, Thomas Norton, and Norman Sadeh. 2019. **Question answering for privacy policies: Combining computational and legal perspectives**. 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 4947–4958, Hong Kong, China. Association for Computational Linguistics.
- Felicia Redelaar, Romy Van Drie, Suzan Verberne, and Maaïke De Boer. 2024. **Attributed question answering for preconditions in the Dutch law**. In *Proceedings of the Natural Legal Language Processing Workshop 2024*, pages 154–165, Miami, FL, USA. Association for 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.
- Stephen Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. pages 0–.
- Francesco Sovrano, Monica Palmirani, Biagio Distanza, Salvatore Sapienza, and Fabio Vitali. 2021. [A dataset for evaluating legal question answering on private international law](#). In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, ICAIL '21*, page 230–234, New York, NY, USA. Association for Computing Machinery.
- Karen Sparck Jones. 1972. [A statistical interpretation of term specificity and its application in retrieval](#). *Journal of Documentation*, 28(1):11–21.
- Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, and Han Xiao. 2025. [Jina embeddings v3: Multilingual text encoder with low-rank adaptations](#). In *Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part V*, page 123–129, Berlin, Heidelberg. Springer-Verlag.
- Weihang Su, Yiran Hu, Anzhe Xie, Qingyao Ai, Quezi Bing, Ning Zheng, Yun Liu, Weixing Shen, and Yiqun Liu. 2024. [STARD: A Chinese statute retrieval dataset derived from real-life queries by non-professionals](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10658–10671, Miami, Florida, USA. Association for Computational Linguistics.
- Xin Sun, Jianan Xie, Zhongqi Chen, Qiang Liu, Shu Wu, Yuehe Chen, Bowen Song, Zilei Wang, Weiqiang Wang, and Liang Wang. 2025. [Divide-then-align: Honest alignment based on the knowledge boundary of rag](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11461–11480.
- 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, and 197 others. 2025a. [Gemma 3 technical report](#). *Preprint*, arXiv:2503.19786.
- GLM Team, Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, Kedong Wang, Lucen Zhong, Mingdao Liu, Rui Lu, Shulin Cao, Xiaohan Zhang, Xuancheng Huang, Yao Wei, and 152 others. 2025b. [Glm-4.5: Agentic, reasoning, and coding \(arc\) foundation models](#). *Preprint*, arXiv:2508.06471.
- Kimi Team, Yifan Bai, Yiping Bao, Y. Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, and 181 others. 2026. [Kimi k2: Open agentic intelligence](#). *Preprint*, arXiv:2507.20534.
- Qwen Team. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Stéphan Tulkens, Chris Emmery, and Walter Daelemans. 2016. [Evaluating unsupervised Dutch word embeddings as a linguistic resource](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4130–4136, Portorož, Slovenia. European Language Resources Association (ELRA).
- Stephan Tulkens and Thomas van Dongen. 2024. [Model2vec: Fast state-of-the-art static embeddings](#).
- Santosh T.y.s.s., Rashid Haddad, and Matthias Grabmair. 2024. [ECtHR-PCR: A dataset for precedent understanding and prior case retrieval in the European court of human rights](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5473–5483, Torino, Italia. ELRA and ICCL.
- VoyageAI. 2024. Domain-specific embeddings and retrieval: Legal edition (voyage-law-2). [VoyageAI blog](#). Accessed: 2025-11-13.
- VoyageAI. 2025. Voyage 3 large. <https://blog.voyageai.com/2025/01/07/voyage-3-large/>. Accessed: 2025-11-13.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. [Text embeddings by weakly-supervised contrastive pre-training](#). *arXiv preprint arXiv:2212.03533*.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024a. [Improving text embeddings with large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024b. [Multilingual e5 text embeddings: A technical report](#). Technical Report MSR-TR-2024-45, Microsoft.
- Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. [MiniLMv2: Multi-head self-attention relation distillation for compressing pre-trained transformers](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics.

Yizhen Wang, Xueying Shen, Zixian Huang, Lihui Niu, and Shiyan Ou. 2024c. [clegal-qa: a chinese legal question answering with natural language generation methods](#). *Complex & Intelligent Systems*, 11.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muenighoff, Defu Lian, and Jian-Yun Nie. 2024. [C-pack: Packed resources for general chinese embeddings](#). In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '24*, page 641–649, New York, NY, USA. Association for Computing Machinery.

Austin Xu, Srijan Bansal, Yifei Ming, Semih Yavuz, and Shafiq Joty. 2025. Does context matter? contextualjudgebench for evaluating llm-based judges in contextual settings. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9541–9564.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, and 9 others. 2025. Qwen2.5-1m technical report. *arXiv preprint arXiv:2501.15383*.

Yixin Yang, Zheng Li, Qingxiu Dong, Heming Xia, and Zhifang Sui. 2024. [Can large multimodal models uncover deep semantics behind images?](#) In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1898–1912, Bangkok, Thailand. Association for Computational Linguistics.

Puxuan Yu, Luke Merrick, Gaurav Nuti, and Daniel Campos. 2024. [Arctic-embed 2.0: Multilingual retrieval without compromise](#). *Preprint*, arXiv:2412.04506.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024a. [mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412, Miami, Florida, US. Association for Computational Linguistics.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, and 1 others. 2024b. [mgte: Generalized long-context text representation and reranking models for multilingual text retrieval](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. [Jec-qa: A legal-domain question answering dataset](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9701–9708.

Youchao Zhou, Heyan Huang, Yicheng Liu, Rui Dai, Xinglin Wang, Xingchen Zhang, Shumin Shi, and Yang Deng. 2026. Do retrieval augmented language models know when they don’t know? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 35158–35166.

## A bLLeQA

Table 3 shows the topic distribution of questions in bLLeQA. Table 4 and Figure 4 show the distribution of articles for the codes that contain relevant instances in the dataset. Figure 5 presents key statistics for the bLLeQA dataset.

Topic	Train	Val	Test	(%)
Housing	327	49	74	30.8
Healthcare	191	37	48	18.9
Family	175	21	14	14.4
Work	114	24	7	9.9
Immigration	122	18	1	9.6
Money	99	13	4	7.9
Privacy	69	12	7	6.0
Justice	28	7	0	2.3

Table 3: Topic distribution of questions in bLLeQA.

## B LLM as a Judge

Table 5 reports the agreement between model predictions and human judgments.

## C Additional Results

Tables 6, 7, 8, 9, 10, 11 and 12 show the detailed results of the retrieval, reranking and RAG experiments.

## D Models

Table 13 presents the models we used in our experiments, as well as their sizes and citations.

## E Prompts

Tables 14, 15 and 16 show the prompt templates used for retrieval, generation and evaluation.

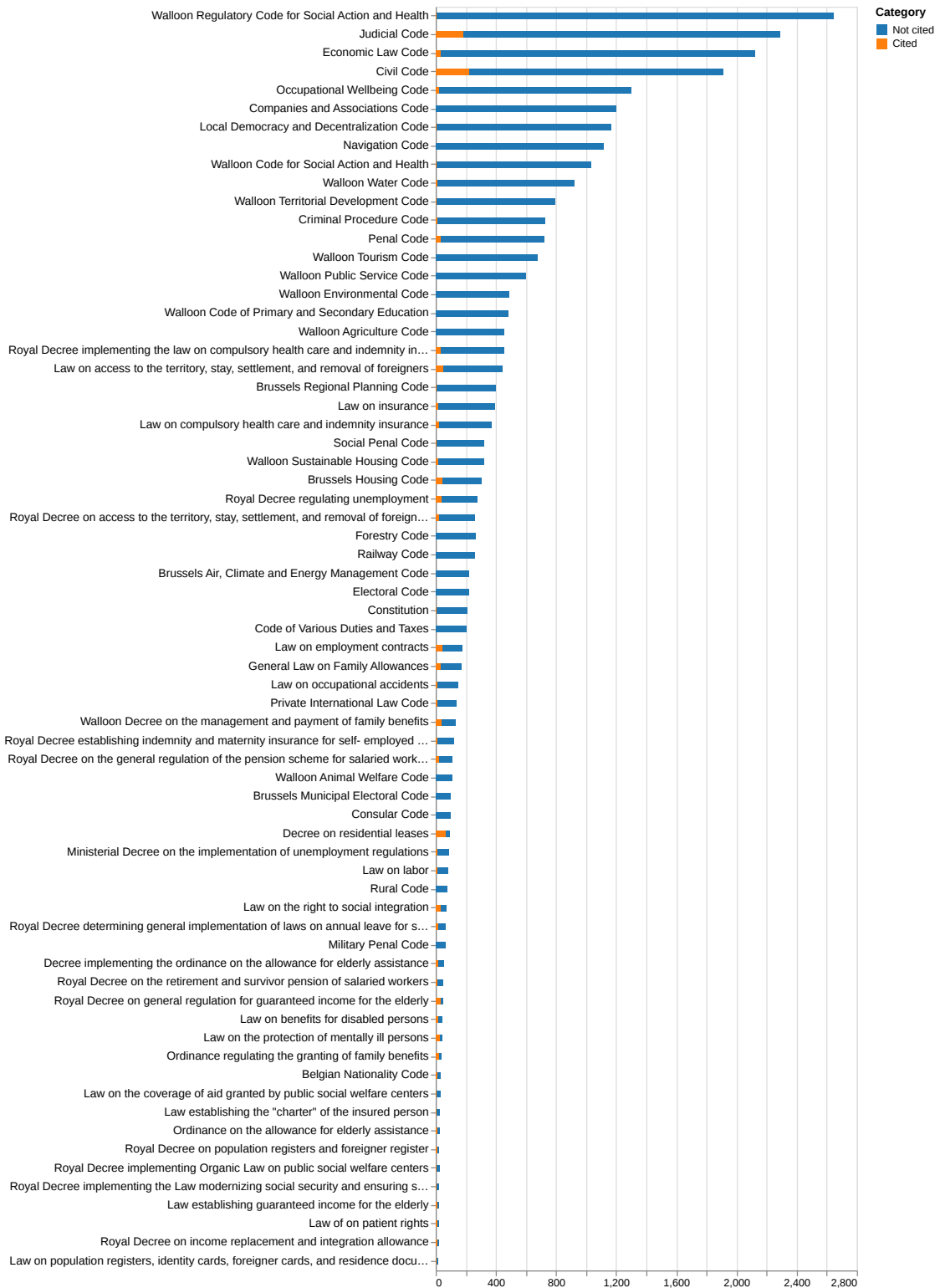


Figure 4: Distribution of codes in the bLLeQA corpus. Articles labeled as “Cited” appear in the training, validation, and test sets, whereas “Not cited” articles do not. Code names are translated for readability.

Code	Total	Relevant
Walloon Regulatory Code for Social Action and Health	2462	1
Judicial Code	2017	156
Economic Law Code	1921	24
Civil Code	1719	158
Code on well-being at work	1270	22
Code of Companies and Associations	1123	0
Code of Local Democracy and Decentralization	1110	3
Walloon Social Action and Health Code	978	3
Belgian Navigation Code	958	0
Environment Code – Water Code – Decree Section	867	7
Walloon Code of Territorial Development	786	4
Penal Code	680	26
Criminal Procedure Code	674	4
Walloon Tourism Code	600	0
Walloon Civil Service Code	490	0
Royal Decree implementing the law on compulsory health insurance and benefits, coordinated on 14 July 1994	448	30
Walloon Agriculture Code	447	0
Law of 15 December 1980 on entry into the territory, residence, establishment, and removal of foreign nationals	440	47
Brussels Code on Spatial Planning	395	1
Insurance Act	390	14
Law on compulsory health insurance and benefits, coordinated on 14 July 1994	369	65
Walloon Code on Sustainable Housing	305	16
Social Penal Code	299	3
Royal Decree regulating unemployment	275	37
Brussels Housing Code	260	36
Railway Code	260	0
Forest Code	259	0
Royal Decree on the entry, stay, settlement and removal of foreign nationals	252	21
Walloon Code on Primary and Secondary Education	238	0
Electoral Code	219	0
Brussels Code on Air, Climate and Energy Management	207	0
The Constitution	206	1
Walloon Environmental Code	198	0
Law on employment contracts	173	40
Environmental Code	172	0
General Law on Family Allowances	168	32
Codes on Miscellaneous Rights and Taxes	162	0
Code on Primary and Secondary Education	151	0
Law on Work Accidents	148	11
Code of Private International Law	132	6
Decree on the management and payment of family benefits	131	35
Royal Decree establishing compensation insurance and maternity insurance for self-employed workers and assisting spouses	120	9
Royal Decree establishing general regulations for the retirement and survivor's pension scheme for salaried workers	110	19
Walloon Animal Welfare Code	106	0
Consular Code	100	0
Brussels Municipal Electoral Code	98	0
Ministerial decree laying down the procedures for implementing unemployment regulations	88	9
Labor Law	85	9
Rural Code	75	0
Royal Decree determining the general terms and conditions for the implementation of laws relating to annual leave for salaried workers	68	13
Law concerning the right to social integration	68	29
Law containing the Military Penal Code	65	0
Decree relating to residential leases	60	44
Royal Decree No. 50 on the Old-Age and Survivors' Pensions for Salaried Workers	52	9
Royal Decree establishing general regulations on income guarantees for elderly persons	49	6
Decree of the Joint Community Commission implementing the Order of 10 December 2020 on allowances for assistance to elderly persons	44	14
Law on allowances for persons with disabilities	43	15
Law on the protection of persons with mental disorders	42	23
Ordinance regulating the granting of family benefits	40	20
Law on the provision of assistance by public social welfare centers	31	5
Belgian Nationality Code	29	1
Environment Code - Book 2: Water Code. - Decree section	27	1
Law establishing the Charter of the Socially Insured Person	25	5
Ordinance on allowances for assistance to the elderly	25	8
Royal Decree of 16 July 1992 on population registers and the register of foreign nationals	24	9
Royal Decree of 9 May 1984 implementing Article 100bis, §1, of the Organic Law of 8 July 1976 on public social assistance centers	24	4
Royal Decree implementing Articles 15, 16 and 17 of the Law of 26 July 1996 on the modernization of social security and ensuring the viability of statutory pension schemes	21	5
Decree of 15 March 2018 on residential leases	19	19
Law establishing income support for the elderly	19	8
Royal Decree on income replacement allowance and integration allowance	18	11
Law on patient rights	15	5
Law of 19 July 1991 on population registers, identity cards, foreigner cards and residence permits	15	3
Decree of the Joint Community Commission of 28 January 2021 implementing the Order of 10 December 2020 on the allowance for assistance to the elderly	10	0
Law of 22 August 2002 on patients' rights	3	2
Royal Decree of 25 November 1991 regulating unemployment	2	0
Royal Decree of 21 December 1967 laying down general regulations for the retirement and survivor's pension scheme for salaried workers	2	2
Decree of the Walloon Region of 8 February 2018 on management and payment	1	1

Table 4: Distribution of legal articles by code. “Total” denotes the number of occurrences in the full corpus, and “Relevant” denotes the number of occurrences among articles referenced across the train/validation/test splits. Code names are translated for readability.

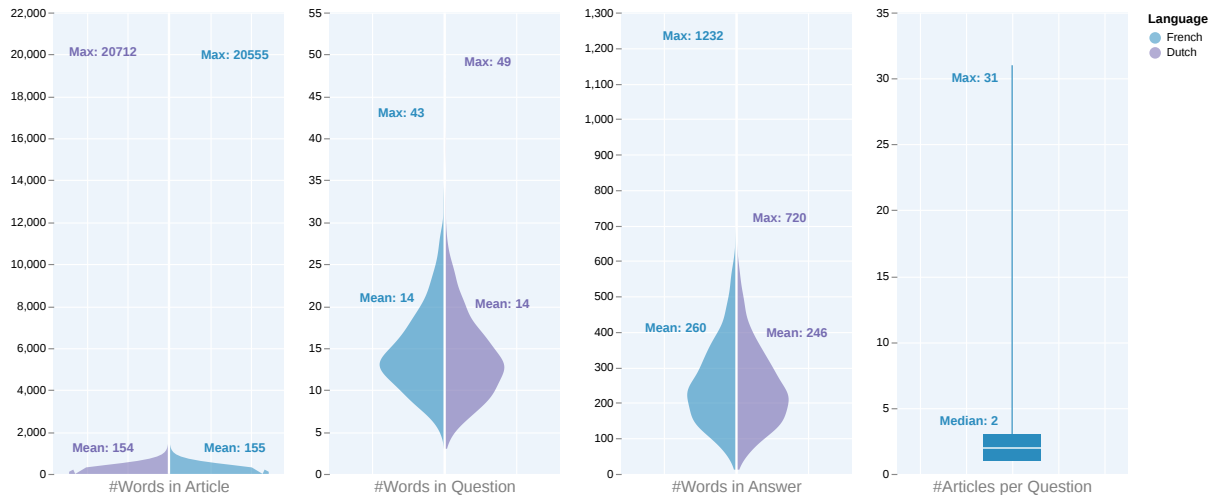


Figure 5: Basic statistics of bLLeQA. From the left: number of words in the articles (French and Dutch), number of words in the questions (French and Dutch), number of words in the answers (French and Dutch), and number of relevant articles per question.

Model	Size	French			Dutch		
		Spearman $\uparrow$	MAE $\downarrow$	F1-macro $\uparrow$	Spearman $\uparrow$	MAE $\downarrow$	F1-macro $\uparrow$
Gemma-3-4B-it	4B	0.49	1.39	49.97	0.30	1.10	55.57
Qwen3.5-9B	9B	0.77	1.30	41.17	0.58	0.93	53.69
Gemma-3-12B-it	12B	0.74	0.95	70.64	0.50	0.92	59.52
GPT-oss-20B	20B	0.71	1.66	37.02	0.46	1.36	49.98
Qwen3.5-27B	27B	0.78	1.15	51.72	<b>0.62</b>	0.89	57.51
Gemma-3-27B-it	27B	0.77	1.31	37.02	0.50	1.01	50.42
GLM-4.7-Flash	30B	0.69	1.51	38.71	0.43	1.25	41.45
Qwen3-30B-A3B-Instruct	30B	0.72	1.61	29.82	0.44	1.46	38.58
Qwen3.5-35B-A3B	35B	0.75	1.42	35.29	0.56	1.03	49.02
Llama-3.3-70B-Instruct	70B	0.71	1.64	36.16	0.54	1.34	51.86
Qwen3-Next-80B-A3B-Instruct	80B	0.67	1.62	37.02	0.44	1.45	46.06
Llama-4-Scout	109B	0.76	1.04	63.74	0.54	0.93	64.63
GPT-oss-120B	120B	0.74	1.27	55.87	0.55	1.02	59.43
Qwen3.5-122B-A10B	122B	0.81	1.37	37.87	0.59	1.00	48.98
Qwen3-235B-A22B	235B	0.73	1.55	27.90	0.49	1.27	42.96
Qwen3.5-397B-A17B	397B	0.76	1.39	37.02	0.55	1.08	49.48
Llama-4-Maverick	400B	0.74	1.07	59.87	0.53	0.89	62.98
Mistral-Large-2512	675B	0.76	1.21	46.62	0.54	0.92	53.53
DeepSeek-v3.2	685B	0.74	1.53	30.77	0.50	1.20	39.29
GLM-5	754B	0.80	1.36	41.17	0.54	1.05	49.98
Kimi-K2-Instruct-0905	1T	0.76	1.39	32.61	0.48	1.22	40.38
Kimi-K2-Thinking	1T	0.69	1.66	26.92	0.46	1.50	33.89
Kimi-K2.5	1.1T	0.79	1.41	31.67	0.56	1.03	41.89
Gemini-3.1-Flash-Lite	–	0.76	1.34	43.55	<b>0.62</b>	1.03	49.98
Claude-Haiku-4.5	–	0.74	1.63	29.82	0.50	1.40	37.04
Gemini-2.5-Flash	–	0.76	1.28	45.86	0.58	1.01	53.69
Gemini-3-Flash	–	0.74	<b>0.83</b>	<b>71.26</b>	0.60	<b>0.71</b>	<b>72.52</b>
Gemini-3.1-Pro	–	0.78	1.30	39.54	0.60	0.97	48.53
Claude-Sonnet-4.6	–	<b>0.82</b>	1.28	37.02	0.59	0.95	48.05

Table 5: Comparison of model judgments against human judgments on DeepSeek’s output for French and Dutch. We report Spearman correlation, MAE, and F1-macro for accurate (scores of 4–5) versus non-accurate (scores of 1–3) answers. Models are ordered by size when available. G-Eval does not support OpenAI models, and Ministral models produced outputs in the wrong format.

Lang	Model	Size	R@100	R@200	R@500	MAP@100	MRR@100	NDCG@10	NDCG@100
FR	TF-IDF	-	59.24	63.11	72.63	18.68	20.73	22.31	27.82
	BM25	-	57.04	64.08	72.50	19.25	22.65	23.94	28.44
	word2vec	-	57.93	66.77	74.56	12.93	16.11	16.42	22.94
	fastText	-	29.78	34.40	45.72	6.70	8.59	9.41	11.99
	static-similarity-mrl-multilingual-v1	-	47.28	55.67	64.89	11.20	13.90	13.69	19.10
	mE5-small	118M	58.44	63.79	71.29	19.09	20.57	23.12	28.10
	potion-multilingual-128M	128M	44.00	55.88	73.56	7.97	9.80	9.21	15.47
	mContriever	178M	54.81	60.80	71.61	9.38	11.72	12.56	19.46
	DPR-XM	277M	38.70	46.98	56.47	8.41	11.29	11.37	15.34
	mE5-base	278M	58.94	64.43	71.95	20.71	22.91	25.37	29.77
	mGTE	305M	65.47	70.74	77.61	20.57	23.17	23.64	30.59
	LaBSE	471M	24.97	33.41	47.90	2.18	3.38	2.52	6.82
	mE5-large	560M	66.84	71.84	77.54	27.80	30.61	32.62	37.31
	mE5-large-instruct	560M	73.46	78.11	84.45	22.49	26.40	26.52	34.30
	BGE-M3	568M	67.30	73.00	78.19	20.45	23.02	25.50	31.58
	snowflake-arctic-embed-l-v2.0	568M	60.61	70.07	81.87	14.62	18.87	17.89	25.18
	jina-embeddings-v3	572M	63.60	71.66	79.61	16.86	20.70	18.80	27.09
	E5-mistral-7b	7B	74.30	77.91	80.70	<b>36.91</b>	41.07	42.61	46.45
	BGE-Mult.-Gemma2	9B	79.86	84.34	88.21	36.22	<b>43.35</b>	<b>42.88</b>	<b>48.01</b>
	voyage-2-law	-	71.44	75.53	82.83	26.69	30.17	31.92	37.61
voyage-3-large	-	<b>81.84</b>	<b>85.93</b>	<b>90.23</b>	34.16	40.65	40.92	46.51	
embedding-3-large	-	79.05	82.59	88.58	33.61	38.56	41.03	45.33	
NL	TF-IDF	-	50.13	57.66	64.86	15.81	17.89	18.62	23.71
	BM25	-	48.52	58.41	63.36	15.46	17.84	17.99	22.93
	word2vec	-	55.37	64.89	75.32	12.02	15.05	14.48	21.48
	fastText	-	44.30	50.40	57.52	9.89	12.97	12.40	17.42
	static-similarity-mrl-multilingual-v1	-	31.39	36.55	48.57	8.46	9.81	9.58	13.59
	E5-small-trm-nl	41M	63.57	67.84	76.88	20.03	22.51	24.12	30.17
	mE5-small	118M	60.64	62.98	72.06	21.87	24.43	25.57	30.85
	E5-base-trm-nl	124M	67.98	72.17	78.30	22.32	25.38	28.18	33.13
	potion-multilingual-128M	128M	44.03	53.81	63.27	8.09	9.38	11.04	16.05
	mContriever	178M	53.89	60.96	76.18	9.92	13.94	14.44	20.00
	DPR-XM	277M	32.44	39.54	49.00	6.58	8.77	8.31	12.38
	mE5-base	278M	61.53	69.16	74.86	21.03	24.20	25.24	30.48
	mGTE	305M	51.77	62.65	69.72	10.81	12.20	12.72	19.69
	E5-large-trm-nl	355M	68.87	74.67	83.37	22.79	26.02	28.56	33.84
	LaBSE	471M	18.62	28.70	48.36	1.96	2.95	2.44	5.45
	mE5-large	560M	68.01	71.82	76.49	25.64	28.95	31.28	35.94
	mE5-large-instruct	560M	71.37	74.56	81.74	26.02	29.46	32.29	36.97
	BGE-M3	568M	65.75	73.97	82.49	20.88	24.97	28.10	32.00
	snowflake-arctic-embed-l-v2.0	568M	62.76	70.91	78.97	13.10	17.33	17.68	24.63
	jina-embeddings-v3	572M	66.39	73.27	79.96	17.05	20.64	21.17	28.26
E5-mistral-7b	7B	74.73	78.35	83.67	<b>36.37</b>	<b>40.69</b>	<b>42.11</b>	<b>46.27</b>	
BGE-Mult.-Gemma2	9B	79.59	83.13	89.72	31.25	36.29	37.73	43.57	
voyage-2-law	-	74.11	78.43	85.22	25.90	30.43	31.87	37.67	
voyage-3-large	-	76.55	81.95	87.46	32.46	37.80	38.88	43.65	
embedding-3-large	-	<b>80.94</b>	<b>85.32</b>	<b>90.52</b>	27.54	32.64	33.08	40.66	

Table 6: Retrieval performance on the French (FR) and Dutch (NL).

Lang	Model	Size	R@10	MAP@10	MAP@100	MRR@10	MRR@100	NDCG@10	NDCG@100	
NL	BM25 +	-	25.69	14.55	15.46	16.88	17.84	17.99	22.93	
	mmarco-mMiniLMv2-L12-H384-v1	0.1B	36.23	19.50	20.21	23.33	23.96	24.57	27.64	
	BGE-reranker-base	0.3B	37.31	19.15	19.80	21.97	22.68	24.33	27.22	
	GTE-multilingual-reranker-base	0.3B	32.15	15.73	16.71	19.23	20.14	20.59	24.65	
	Jina-reranker-v2-base-multilingual	0.3B	35.67	18.56	19.31	22.06	22.65	23.62	26.82	
	BGE-reranker-large	0.6B	38.76	21.81	22.38	25.94	26.42	27.07	29.50	
	BGE-reranker-v2-m3	0.6B	<b>39.11</b>	<b>22.34</b>	<b>22.92</b>	<b>26.96</b>	<b>27.43</b>	<b>27.54</b>	<b>29.93</b>	
	mE5-small +	0.1B	36.98	20.87	21.87	23.41	24.43	25.57	30.85	
	mmarco-mMiniLMv2-L12-H384-v1	0.1B	41.93	18.46	19.38	21.44	22.40	24.81	29.23	
	BGE-reranker-base	0.3B	34.62	14.05	15.45	16.84	18.23	19.69	25.92	
	GTE-multilingual-reranker-base	0.3B	33.44	15.47	16.91	18.00	19.54	20.47	26.93	
	Jina-reranker-v2-base-multilingual	0.3B	34.51	16.80	18.13	19.70	21.04	21.90	28.01	
	BGE-reranker-large	0.6B	43.84	20.51	21.41	24.30	25.15	27.24	31.20	
	BGE-reranker-v2-m3	0.6B	<b>44.51</b>	<b>23.10</b>	<b>23.87</b>	<b>27.64</b>	<b>28.44</b>	<b>29.42</b>	<b>33.12</b>	
	mE5-base +	0.2B	37.95	69.16	20.06	21.03	24.20	25.24	30.48	
	mmarco-mMiniLMv2-L12-H384-v1	0.1B	41.69	20.18	21.22	23.27	24.21	26.14	30.84	
	BGE-reranker-base	0.3B	37.39	15.95	17.25	18.85	19.92	21.84	27.53	
	GTE-multilingual-reranker-base	0.3B	38.68	16.81	17.98	19.72	20.76	22.76	28.05	
	Jina-reranker-v2-base-multilingual	0.3B	37.71	17.33	18.65	19.66	20.87	22.94	28.64	
	BGE-reranker-large	0.6B	45.53	21.52	22.42	25.34	26.09	28.36	32.18	
	BGE-reranker-v2-m3	0.6B	<b>47.25</b>	<b>24.44</b>	<b>25.13</b>	<b>29.48</b>	<b>30.05</b>	<b>31.21</b>	<b>34.46</b>	
	mE5-large-instruct +	0.5B	50.83	25.06	26.02	28.66	29.46	32.29	36.97	
	mmarco-mMiniLMv2-L12-H384-v1	0.1B	43.73	20.66	21.89	23.99	25.18	27.06	33.33	
	BGE-reranker-base	0.3B	34.59	15.06	16.87	17.98	19.76	20.43	28.99	
	GTE-multilingual-reranker-base	0.3B	35.34	15.43	17.26	18.23	19.99	20.99	29.36	
	Jina-reranker-v2-base-multilingual	0.3B	46.74	19.90	21.12	23.09	24.20	27.23	32.98	
	BGE-reranker-large	0.6B	46.29	20.91	22.29	25.55	26.90	28.17	34.21	
	BGE-reranker-v2-m3	0.6B	<b>51.88</b>	<b>25.56</b>	<b>26.42</b>	<b>30.10</b>	<b>30.89</b>	<b>33.14</b>	<b>37.41</b>	
	voyage-3-large +	-	<b>56.15</b>	<b>31.39</b>	<b>32.46</b>	<b>37.10</b>	<b>37.80</b>	<b>38.88</b>	<b>43.65</b>	
	mmarco-mMiniLMv2-L12-H384-v1	0.1B	48.57	22.00	23.43	25.73	26.88	29.46	36.11	
	BGE-reranker-base	0.3B	39.22	16.50	18.47	19.88	21.58	22.77	31.67	
	GTE-multilingual-reranker-base	0.3B	38.14	16.58	18.59	20.38	22.07	22.73	31.78	
	Jina-reranker-v2-base-multilingual	0.3B	48.14	21.54	23.03	24.88	26.09	28.93	35.70	
	BGE-reranker-large	0.6B	43.49	18.74	20.68	23.29	24.83	26.94	33.95	
	BGE-reranker-v2-m3	0.6B	51.96	26.80	28.10	32.02	32.99	34.30	40.04	
	FR	BM25 +	-	37.74	18.36	19.25	21.84	22.65	23.94	28.44
		mmarco-mMiniLMv2-L12-H384-v1	0.1B	39.54	19.05	20.00	21.86	22.85	24.76	29.16
		BGE-reranker-base	0.3B	30.96	12.88	14.26	15.93	17.20	18.19	24.39
		GTE-multilingual-reranker-base	0.3B	41.29	22.49	23.42	25.57	26.44	28.01	31.97
		Jina-reranker-v2-base-multilingual	0.3B	37.84	12.84	13.71	15.78	16.60	19.57	23.99
		BGE-reranker-large	0.6B	36.77	21.10	22.33	25.41	26.67	25.99	31.12
		BGE-reranker-v2-m3	0.6B	<b>43.95</b>	<b>26.18</b>	<b>27.01</b>	<b>31.46</b>	<b>32.09</b>	<b>31.92</b>	<b>35.24</b>
		mE5-small +	0.1B	37.41	18.07	19.09	19.42	20.57	23.12	28.10
		mmarco-mMiniLMv2-L12-H384-v1	0.1B	39.65	20.07	20.89	23.60	24.47	25.65	29.92
		BGE-reranker-base	0.3B	26.23	10.41	11.86	11.95	13.60	14.63	21.93
		GTE-multilingual-reranker-base	0.3B	41.58	21.95	22.80	24.97	25.87	27.58	31.57
		Jina-reranker-v2-base-multilingual	0.3B	34.70	12.16	13.33	15.32	16.42	18.37	23.86
		BGE-reranker-large	0.6B	39.86	18.54	19.48	21.66	22.53	24.45	28.74
BGE-reranker-v2-m3		0.6B	<b>44.27</b>	<b>25.83</b>	<b>26.53</b>	<b>29.78</b>	<b>30.40</b>	<b>31.38</b>	<b>34.63</b>	
mE5-base +		0.2B	40.00	19.82	20.71	21.91	22.91	25.37	29.77	
mmarco-mMiniLMv2-L12-H384-v1		0.1B	39.97	20.46	21.33	23.41	24.28	25.94	30.31	
BGE-reranker-base		0.3B	30.12	11.73	12.97	13.37	14.80	16.59	23.09	
GTE-multilingual-reranker-base		0.3B	42.66	22.65	23.43	25.19	25.89	28.31	32.05	
Jina-reranker-v2-base-multilingual		0.3B	36.61	12.54	13.66	15.40	16.45	18.99	24.20	
BGE-reranker-large		0.6B	39.75	20.24	21.19	23.41	24.27	25.73	30.15	
BGE-reranker-v2-m3		0.6B	<b>45.00</b>	<b>26.56</b>	<b>27.26</b>	<b>30.61</b>	<b>31.24</b>	<b>32.13</b>	<b>35.36</b>	
mE5-large-instruct +		0.5B	44.56	20.61	21.92	23.83	24.97	27.29	33.87	
mmarco-mMiniLMv2-L12-H384-v1		0.1B	44.16	20.22	21.53	23.11	24.41	26.77	33.59	
BGE-reranker-base		0.3B	27.63	11.15	13.18	12.66	14.74	15.55	25.99	
GTE-multilingual-reranker-base		0.3B	45.67	22.62	23.98	25.06	26.40	29.07	35.66	
Jina-reranker-v2-base-multilingual		0.3B	40.29	13.68	15.33	16.53	18.07	20.81	28.68	
BGE-reranker-large		0.6B	41.69	20.63	22.10	24.37	25.77	26.68	33.99	
BGE-reranker-v2-m3		0.6B	<b>49.62</b>	<b>29.75</b>	<b>30.98</b>	<b>34.15</b>	<b>35.26</b>	<b>35.84</b>	<b>41.53</b>	
voyage-3-large +		-	<b>58.52</b>	<b>32.82</b>	<b>34.16</b>	<b>39.72</b>	<b>40.65</b>	<b>40.92</b>	<b>46.51</b>	
mmarco-mMiniLMv2-L12-H384-v1		0.1B	46.96	24.14	25.86	27.55	29.11	30.64	38.90	
BGE-reranker-base		0.3B	30.66	10.76	12.92	12.42	14.52	16.09	27.49	
GTE-multilingual-reranker-base		0.3B	47.95	22.77	24.16	24.95	26.21	29.63	37.18	
Jina-reranker-v2-base-multilingual		0.3B	40.59	13.90	15.85	16.91	18.59	21.14	30.62	
BGE-reranker-large		0.6B	41.37	19.49	21.21	23.22	24.64	25.70	34.65	
BGE-reranker-v2-m3		0.6B	51.98	30.15	31.65	35.10	36.23	36.77	43.73	

Table 7: Reranker performance on retrieved results from the Dutch (NL) and French (FR) subsets of bLLeQA (test set). First row of each block: retrieval-only baseline; subsequent rows: rerankers applied to the retriever results above.

Model	Size	Setting	French			Dutch		
			Pr	Rec	F1	Pr	Rec	F1
Ministral-3B	3B	Gold	81.94	72.56	75.55	80.00	71.44	74.22
		RAG+	23.17	27.69	23.52	22.46	29.35	23.51
		RAG	32.22	37.74	33.20	37.60	44.84	38.36
Gemma-3-4B-it	4B	Gold	83.31	73.40	76.12	69.68	64.60	66.26
		RAG+	7.53	39.30	10.85	7.06	37.47	10.14
		RAG	6.89	34.60	9.68	6.75	30.81	9.53
Ministral-8B	8B	Gold	98.71	89.85	92.71	98.71	88.99	92.16
		RAG+	39.03	57.66	42.00	42.91	54.33	44.39
		RAG	37.91	53.95	40.42	38.13	49.35	40.27
Qwen3.5-9B	9B	Gold	76.77	68.77	71.21	73.55	65.70	68.24
		RAG+	43.52	59.22	46.31	42.85	55.66	44.70
		RAG	47.33	58.82	49.37	52.19	61.45	53.58
Gemma-3-12B-it	12B	Gold	95.27	84.20	87.77	91.83	80.45	84.04
		RAG+	18.07	33.63	20.34	15.84	27.31	17.05
		RAG	15.34	29.46	17.04	19.97	33.44	22.36
Ministral-14B	14B	Gold	97.58	88.97	91.77	97.63	87.51	90.68
		RAG+	44.34	56.51	45.28	46.32	53.95	46.01
		RAG	43.61	55.13	45.36	44.46	51.34	44.20
GPT-oss-20B	20B	Gold	80.97	72.81	75.19	74.19	65.58	68.26
		RAG+	33.30	36.24	32.56	33.89	37.50	33.19
		RAG	34.61	36.53	33.77	39.53	42.34	38.45
Gemma-3-27B-it	27B	Gold	85.16	75.55	78.65	89.35	78.13	81.58
		RAG+	28.48	32.37	27.92	26.08	28.17	25.53
		RAG	27.41	30.75	27.14	27.28	27.63	26.39
Qwen3.5-27B	27B	Gold	77.42	70.41	72.73	77.42	70.20	72.45
		RAG+	47.19	72.44	52.54	48.37	73.78	54.00
		RAG	49.37	69.17	53.58	52.02	70.89	56.21
GLM-4.7-Flash	30B	Gold	96.62	89.63	91.57	93.87	88.12	90.04
		RAG+	24.79	58.06	29.81	25.18	62.47	31.06
		RAG	24.20	56.99	29.70	22.77	52.07	27.15
Qwen3-30B-A3B-Instruct	30B	Gold	96.77	84.93	88.68	98.71	88.11	91.46
		RAG+	40.74	54.70	42.29	36.68	57.02	39.33
		RAG	39.46	53.92	41.28	36.97	57.10	40.26
Qwen3.5-35B-A3B	35B	Gold	71.61	66.54	68.15	74.19	68.31	70.19
		RAG+	50.33	64.99	52.91	48.45	62.84	50.71
		RAG	54.98	67.77	57.19	54.69	65.22	56.34
Llama-3.3-70B-Instruct	70B	Gold	90.81	79.37	82.92	90.97	81.27	84.45
		RAG+	36.03	45.38	36.33	35.45	46.72	35.72
		RAG	43.31	50.22	43.24	38.25	44.84	37.68
Qwen3-Next-80B-A3B-Instruct	80B	Gold	89.68	78.46	81.79	91.61	79.67	83.28
		RAG+	46.94	61.15	47.53	44.43	61.56	46.25
		RAG	48.27	58.66	48.24	49.32	60.81	49.88
Llama-4-Scout	109B	Gold	90.32	80.68	83.75	88.17	78.93	81.90
		RAG+	36.73	47.01	37.35	33.82	44.73	34.75
		RAG	39.75	48.25	40.61	36.13	47.02	37.51
GPT-oss-120B	120B	Gold	85.48	76.92	79.65	85.16	76.57	79.24
		RAG+	39.81	50.30	41.25	38.89	47.07	40.06
		RAG	41.49	48.79	42.61	40.09	47.61	41.11
Qwen3.5-122B-A10B	122B	Gold	70.97	63.48	65.94	65.16	59.35	61.27
		RAG+	52.86	71.96	56.60	50.11	64.46	52.35
		RAG	55.94	69.81	58.79	58.20	68.31	60.05
Qwen3-235B-A22B	235B	Gold	95.48	84.53	87.89	93.55	82.95	86.25
		RAG+	48.17	64.60	49.91	48.42	59.65	49.31
		RAG	45.57	59.95	47.33	43.86	53.82	44.30
Qwen3.5-397B-A17B	397B	Gold	72.26	65.42	67.66	74.19	65.42	68.08
		RAG+	50.22	69.59	54.03	50.98	70.05	54.80
		RAG	59.49	76.13	62.39	54.62	66.18	55.88
Llama-4-Maverick	400B	Gold	88.82	78.37	81.72	89.68	79.15	82.25
		RAG+	46.37	55.40	46.60	48.25	63.17	50.01
		RAG	49.64	62.15	51.20	51.34	56.94	50.43
Mistral-Large-2512	675B	Gold	95.48	86.95	89.69	93.55	85.12	87.77
		RAG+	50.19	68.36	53.52	52.38	68.95	55.00
		RAG	46.13	62.37	49.15	47.32	61.77	49.56
DeepSeek-v3.2	685B	Gold	76.77	69.25	71.60	90.19	81.04	83.97
		RAG+	52.89	68.99	55.43	49.52	69.29	52.94
		RAG	53.50	70.24	56.01	48.55	62.63	49.89
GLM-5	754B	Gold	76.13	68.78	71.13	75.48	68.16	70.49
		RAG+	55.52	74.59	58.89	56.49	69.38	58.49
		RAG	59.55	71.21	61.03	58.06	68.41	59.23
Kimi-K2-Instruct-0905	1T	Gold	85.16	72.87	76.78	86.29	75.42	78.61
		RAG+	46.46	56.10	47.20	38.49	46.85	38.83
		RAG	48.95	53.49	48.09	40.00	48.25	40.42
Kimi-K2-Thinking	1T	Gold	64.52	58.48	60.38	66.45	59.53	61.69
		RAG+	46.66	69.72	50.81	43.83	61.51	47.26
		RAG	53.10	72.02	56.82	51.40	68.82	54.70
Kimi-K2.5	1.1T	Gold	69.68	62.03	64.47	66.45	60.56	62.39
		RAG+	54.62	63.59	55.44	54.65	65.19	55.58
		RAG	64.37	71.40	64.49	62.55	68.92	62.55
GPT-5-Nano	-	Gold	86.45	79.57	81.90	87.10	80.79	82.75
		RAG+	33.04	44.09	34.27	28.04	38.03	29.26
		RAG	35.82	44.44	36.18	36.26	45.46	37.06
Gemini-3.1-Flash-Lite	-	Gold	78.71	70.74	73.21	78.06	70.34	72.66
		RAG+	64.62	69.30	63.58	65.64	67.49	63.26
		RAG	63.74	67.04	62.45	63.75	65.65	61.77
GPT-5-Mini	-	Gold	81.29	79.52	80.18	82.58	79.84	80.80
		RAG+	35.74	70.56	43.34	33.93	73.19	42.21
		RAG	32.94	62.12	39.52	37.21	67.66	43.24
Gemini-2.5-Flash	-	Gold	79.03	70.82	73.49	81.29	74.19	76.43
		RAG+	46.94	72.67	51.59	43.28	72.19	49.41
		RAG	46.94	68.84	51.18	41.56	67.74	47.04
Gemini-3-Flash	-	Gold	80.65	76.11	77.49	83.87	79.39	80.94
		RAG+	42.49	76.76	50.31	43.21	78.05	51.74
		RAG	44.25	72.80	50.46	40.16	70.78	46.92
Claude-Haiku-4.5	-	Gold	77.42	69.33	71.97	76.77	68.68	71.25
		RAG+	50.18	71.74	53.78	53.53	67.66	55.87
		RAG	54.05	69.57	56.21	55.62	66.05	56.30
Gemini-3.1-Pro	-	Gold	78.71	72.03	74.17	79.35	71.66	74.13
		RAG+	60.88	69.96	61.51	62.85	71.19	63.19
		RAG	63.88	71.56	63.88	64.28	69.73	63.43
GPT-5.4	-	Gold	70.32	65.74	67.19	70.97	65.59	67.35
		RAG+	41.86	71.63	48.57	41.73	67.33	47.03
		RAG	52.22	75.38	57.49	52.13	74.73	56.73
Claude-Sonnet-4.6	-	Gold	86.45	83.07	84.22	88.39	83.72	85.27
		RAG+	43.60	82.19	52.10	44.88	78.40	52.25
		RAG	42.64	71.26	48.72	41.82	71.61	48.43

Table 8: Citation coverage precision (Pr), recall (Rec), and F1 for Dutch and French, ordered by model size when available.

model	size	setting	French		Dutch	
			Correctness	Faithfulness	Correctness	Faithfulness
Ministral-3B	3B	Gold	2.7226	0.7429	2.6258	0.6827
		RAG+	2.3032	0.6506	2.2774	0.6434
		RAG	2.6323	0.6270	2.7226	0.6094
Gemma-3-4B-it	4B	Gold	2.6645	0.7820	2.2258	0.6944
		RAG+	2.0065	0.4666	1.9355	0.4920
		RAG	2.1226	0.4618	1.8710	0.4938
Ministral-8B	8B	Gold	3.3613	0.7506	3.2839	0.7624
		RAG+	3.4129	0.6930	3.2387	0.7409
		RAG	3.2516	0.7346	3.1871	0.7090
Qwen3.5-9B	9B	Gold	2.8323	0.8900	2.6903	0.9257
		RAG+	3.1226	0.8828	3.0387	0.8661
		RAG	3.3226	0.9004	3.2129	0.8454
Gemma-3-12B-it	12B	Gold	3.0581	0.9013	2.9548	0.8668
		RAG+	2.2194	0.6296	2.0194	0.6462
		RAG	2.2194	0.6292	2.2323	0.6907
Ministral-14b	14B	Gold	3.5290	0.7977	3.4000	0.7969
		RAG+	3.3290	0.7269	3.1548	0.7318
		RAG	3.4387	0.7692	3.1806	0.7486
GPT-oss-20B	20B	Gold	2.8774	0.8646	2.7355	0.7870
		RAG+	2.6581	0.6566	2.3548	0.6552
		RAG	2.8323	0.6675	2.7548	0.6586
Gemma-3-27B-it	27B	Gold	2.8903	0.9067	2.8903	0.9203
		RAG+	2.3871	0.8501	2.1806	0.8190
		RAG	2.3935	0.8384	2.2839	0.8161
Qwen3.5-27B	27B	Gold	3.0258	0.9168	2.8968	0.9354
		RAG+	3.6516	0.9388	3.5097	0.9510
		RAG	3.7032	0.9692	3.6452	0.9344
Qwen3-30B-A3B-Instruct	30B	Gold	3.3226	0.8679	3.4323	0.8264
		RAG+	3.0980	0.7969	2.9605	0.7142
		RAG	3.2013	0.7828	3.0395	0.7673
GLM-4.7-Flash	30B	Gold	3.2129	0.8561	3.0129	0.8430
		RAG+	2.8839	0.7826	2.7613	0.7523
		RAG	2.8516	0.7854	2.7613	0.7608
Qwen3.5-35B-A3B	35B	Gold	2.8839	0.8832	2.8645	0.9485
		RAG+	3.3871	0.9411	3.2323	0.9064
		RAG	3.5742	0.9445	3.5226	0.9013
Llama-3.3-70B-Instruct	70B	Gold	2.6194	0.8874	2.8774	0.8471
		RAG+	2.3613	0.7758	2.2387	0.6843
		RAG	2.4774	0.7978	2.5226	0.7033
Qwen3-Next-80B-A3B-Instruct	80B	Gold	3.1226	0.9053	2.9742	0.8811
		RAG+	3.2774	0.8937	2.9677	0.8304
		RAG	3.2581	0.8652	3.1097	0.8098
Llama-4-Scout	109B	Gold	2.8774	0.8882	2.9032	0.8594
		RAG+	2.4645	0.7488	2.4774	0.6886
		RAG	2.6516	0.7794	2.5548	0.6944
GPT-oss-120B	120B	Gold	3.0452	0.8775	2.9613	0.8825
		RAG+	3.0323	0.7868	2.8000	0.7731
		RAG	3.1548	0.7558	3.1032	0.7764
Qwen3.5-122B-A10B	122B	Gold	2.8903	0.9599	2.7419	0.9380
		RAG+	3.4710	0.9591	3.3548	0.8893
		RAG	3.7161	0.9443	3.6452	0.9419
Qwen3-235B-A22B	235B	Gold	3.4774	0.9083	3.3806	0.8752
		RAG+	3.2323	0.8634	3.1613	0.7978
		RAG	3.1677	0.8647	3.0581	0.8294
Qwen3.5-397B-A17B	397B	Gold	2.9613	0.9071	2.9419	0.9302
		RAG+	3.5806	0.9247	3.5161	0.9485
		RAG	3.8839	0.9671	3.7097	0.9326
Llama-4-Maverick	400B	Gold	3.0645	0.9164	2.8774	0.8798
		RAG+	2.8065	0.8505	2.9548	0.8180
		RAG	3.0974	0.8467	3.0323	0.8160
Mistral-Large-2512	675B	Gold	3.5677	0.8749	3.5419	0.8500
		RAG+	3.7806	0.8549	3.7161	0.8356
		RAG	3.6774	0.8504	3.6774	0.8224
DeepSeek-v3.2	685B	Gold	2.9806	0.8956	3.1161	0.9068
		RAG+	3.4581	0.9288	3.3806	0.8781
		RAG	3.7226	0.9291	3.2903	0.9157
GLM-5	754B	Gold	3.1484	0.9002	3.0774	0.9430
		RAG+	3.6194	0.9593	3.4968	0.9394
		RAG	3.8129	0.9499	3.7161	0.9273
Kimi-K2-Instruct-0905	1T	Gold	3.0714	0.8586	3.0258	0.7778
		RAG+	3.2387	0.7515	2.8323	0.6869
		RAG	3.1935	0.7582	2.9226	0.6774
Kimi-K2-Thinking	1T	Gold	2.9290	0.8520	2.9032	0.9635
		RAG+	3.5548	0.8950	3.3355	0.9009
		RAG	3.8710	0.9278	3.7226	0.8885
Kimi-K2.5	1.1T	Gold	3.0065	0.9024	2.9032	0.9441
		RAG+	3.4258	0.9106	3.3613	0.9313
		RAG	3.8387	0.9380	3.7290	0.9300
GPT-5-Nano	-	Gold	3.2710	0.9088	3.2774	0.8873
		RAG+	3.2000	0.7548	3.0581	0.7276
		RAG	3.2645	0.7894	3.2065	0.7732
Gemini-3.1-Flash-Lite	-	Gold	3.1097	0.9233	3.0516	0.9114
		RAG+	3.4065	0.9466	3.3742	0.8943
		RAG	3.5484	0.9398	3.4581	0.8914
GPT-5-Mini	-	Gold	3.3548	0.9355	3.3161	0.9466
		RAG+	3.5548	0.9547	3.6903	0.9534
		RAG	3.6710	0.9540	3.7032	0.9388
Gemini-2.5-Flash	-	Gold	3.1806	0.9175	3.2581	0.9223
		RAG+	3.7226	0.8942	3.7806	0.9046
		RAG	3.7742	0.9342	3.6710	0.9180
Gemini-3-Flash	-	Gold	3.3355	0.9192	3.3677	0.9536
		RAG+	3.9161	0.9719	3.8387	0.9667
		RAG	3.9935	0.9814	3.8452	0.9573
Claude-Haiku-4.5	-	Gold	3.1548	0.8949	3.1032	0.9446
		RAG+	3.5484	0.9502	3.4065	0.9265
		RAG	3.7742	0.9017	3.6000	0.9010
Gemini-3.1-Pro	-	Gold	3.1871	0.9391	3.1613	0.9416
		RAG+	3.6065	0.9661	3.4645	0.9475
		RAG	3.7355	0.9683	3.7032	0.9220
GPT-5.4	-	Gold	3.1226	0.8907	3.1613	0.9555
		RAG+	3.6194	0.9621	3.5419	0.9515
		RAG	4.0323	0.9475	3.9548	0.9482
Claude-Sonnet-4.6	-	Gold	3.6323	0.9248	3.6129	0.9488
		RAG+	4.1355	0.9714	4.0387	0.9480
		RAG	4.1548	0.9621	4.1290	0.9455

Table 9: Answer correctness and faithfulness results by model, setting, and language.

model	size	setting	French				Dutch			
			AccAns	InaccAns	CorrRef	IncorrRef	AccAns	InaccAns	CorrRef	IncorrRef
Ministral-3B	3B	Gold	27.74	56.77	0	15.48	29.03	53.55	0	17.42
		RAG+	21.94	44.52	0	33.55	20.00	50.32	0	29.68
		RAG	17.42	45.16	11.61	25.81	20.00	47.74	12.26	20.00
Gemma-3-4B-it	4B	Gold	23.87	67.74	0	8.39	14.19	69.68	0	16.13
		RAG+	7.74	89.68	0	2.58	7.10	90.97	0	1.94
		RAG	10.97	86.45	1.94	0.65	3.23	91.61	0.65	4.52
Ministral-8B	8B	Gold	52.26	47.74	0	0	45.16	54.84	0	0
		RAG+	52.90	47.10	0	0	49.03	49.03	0	1.94
		RAG	47.10	52.90	0	0	43.23	55.48	0.65	0.65
Qwen3.5-9B	9B	Gold	32.90	44.52	0	22.58	27.74	47.10	0	25.16
		RAG+	41.29	51.61	0	7.10	35.48	55.48	0	9.03
		RAG	37.42	50.32	7.10	5.16	32.26	54.19	7.74	5.81
Gemma-3-12B-it	12B	Gold	31.61	67.74	0	0.65	30.32	65.16	0	4.52
		RAG+	16.77	72.90	0	10.32	10.32	67.74	0	21.94
		RAG	11.61	78.71	1.94	7.74	10.32	67.74	5.81	16.13
Ministral-14b	14B	Gold	54.19	44.52	0	1.29	50.32	47.74	0	1.94
		RAG+	51.61	44.52	0	3.87	47.10	44.52	0	8.39
		RAG	50.32	46.45	2.58	0.65	43.23	50.32	1.94	4.52
GPT-oss-20B	20B	Gold	29.68	52.26	0	18.06	30.32	47.10	0	22.58
		RAG+	27.10	57.42	0	15.48	18.71	59.35	0	21.94
		RAG	23.87	60.00	5.16	10.97	19.35	63.23	6.45	10.97
Qwen3.5-27B	27B	Gold	41.94	35.48	0	22.58	36.13	37.42	0	26.45
		RAG+	61.29	32.26	0	6.45	56.77	35.48	0	7.74
		RAG	56.13	36.13	5.16	2.58	50.97	37.42	5.16	3.87
Gemma-3-27B-it	27B	Gold	29.68	58.06	0	12.26	27.74	62.58	0	9.68
		RAG+	16.77	76.13	0	7.10	11.61	82.58	0	5.81
		RAG	14.84	78.71	3.23	3.23	11.61	83.87	1.94	2.58
Qwen3-30B-A3B-Instruct	30B	Gold	46.45	50.32	0	3.23	51.61	47.10	0	1.29
		RAG+	39.22	57.52	0	3.27	32.24	63.82	0	3.95
		RAG	40.91	55.19	1.95	1.95	37.50	60.53	0.66	1.32
GLM-4.7-Flash	30B	Gold	36.13	63.87	0	0	36.77	60.00	0	3.23
		RAG+	29.03	70.97	0	0	25.16	74.84	0	0
		RAG	31.61	68.39	0	0	21.94	76.13	1.29	0.65
Qwen3.5-35B-A3B	35B	Gold	37.42	35.48	0	27.10	38.71	36.13	0	25.16
		RAG+	54.19	36.13	0	9.68	45.81	41.94	0	12.26
		RAG	49.68	37.42	8.39	4.52	44.52	39.35	9.68	6.45
Llama-3.3-70B-Instruct	70B	Gold	17.42	73.55	0	9.03	26.45	64.52	0	9.03
		RAG+	10.97	78.06	0	10.97	7.10	80.00	0	12.90
		RAG	7.74	76.13	7.74	8.39	13.55	76.77	3.87	5.81
Qwen3-Next-80B-A3B-Instruct	80B	Gold	37.42	53.55	0	9.03	32.90	58.71	0	8.39
		RAG+	46.45	49.03	0	4.52	31.61	63.87	0	4.52
		RAG	38.06	56.77	3.23	1.94	30.97	64.52	3.87	0.65
Llama-4-Scout	109B	Gold	23.87	67.10	0	9.03	32.90	56.13	0	10.97
		RAG+	16.13	65.81	0	18.06	15.48	78.06	0	6.45
		RAG	17.42	69.03	3.87	9.68	13.55	80.65	1.29	4.52
GPT-oss-120B	120B	Gold	34.84	50.97	0	14.19	33.55	51.61	0	14.84
		RAG+	34.19	58.71	0	7.10	27.74	62.58	0	9.68
		RAG	36.77	52.90	4.52	5.81	34.19	54.84	5.81	5.16
Qwen3.5-122B-A10B	122B	Gold	37.42	34.19	0	28.39	36.77	28.39	0	34.84
		RAG+	52.90	38.71	0	8.39	50.97	38.06	0	10.97
		RAG	50.32	40.00	8.39	1.29	50.32	32.90	9.68	7.10
Qwen3-235B-A22B	235B	Gold	54.84	40.65	0	4.52	50.32	43.23	0	6.45
		RAG+	47.10	49.68	0	3.23	40.00	56.77	0	3.23
		RAG	40.00	54.84	2.58	2.58	34.19	60.65	2.58	2.58
Qwen3.5-397B-A17B	397B	Gold	41.94	30.32	0	27.74	38.06	36.77	0	25.16
		RAG+	60.65	29.68	0	9.68	56.77	31.61	0	11.61
		RAG	57.42	31.61	9.03	1.94	52.26	33.55	8.39	5.81
Llama-4-Maverick	400B	Gold	32.90	56.77	0	10.32	26.45	63.23	0	10.32
		RAG+	27.74	58.71	0	13.55	29.03	67.10	0	3.87
		RAG	29.87	57.79	6.49	5.84	27.74	69.03	2.58	0.65
Mistral-Large-2512	675B	Gold	57.42	38.71	0	3.87	58.71	34.84	0	6.45
		RAG+	67.74	32.26	0	0	63.23	36.77	0	0
		RAG	60.65	39.35	0	0	66.45	33.55	0	0
DeepSeek-v3.2	685B	Gold	43.23	33.55	0	23.23	38.06	52.26	0	9.68
		RAG+	56.77	31.61	0	11.61	49.03	46.45	0	4.52
		RAG	50.97	34.84	9.03	5.16	42.58	53.55	2.58	1.29
GLM-5	754B	Gold	48.39	27.74	0	23.87	44.52	31.61	0	23.87
		RAG+	60.00	33.55	0	6.45	56.13	35.48	0	8.39
		RAG	52.90	34.84	9.03	3.23	54.19	36.77	5.81	3.23
Kimi-K2-Instruct-0905	1T	Gold	38.31	53.90	0	7.79	36.13	59.35	0	4.52
		RAG+	43.87	52.90	0	3.23	30.97	58.71	0	10.32
		RAG	40.65	53.55	3.87	1.94	37.42	57.42	1.29	3.87
Kimi-K2-Thinking	1T	Gold	47.74	18.71	0	33.55	44.52	21.94	0	33.55
		RAG+	60.65	27.74	0	11.61	51.61	33.55	0	14.84
		RAG	61.29	25.16	9.68	3.87	52.90	33.55	9.03	4.52
Kimi-K2.5	1.1T	Gold	44.52	25.81	0	29.68	43.23	23.87	0	32.90
		RAG+	56.13	27.74	0	16.13	52.90	29.03	0	18.06
		RAG	58.71	23.23	12.26	5.81	45.81	35.48	12.26	6.45
GPT-5-Nano	-	Gold	49.68	36.77	0	13.55	46.45	40.65	0	12.90
		RAG+	47.74	41.29	0	10.97	41.94	47.10	0	10.97
		RAG	43.87	43.87	3.87	8.39	41.94	45.81	4.52	7.74
Gemini-3.1-Flash-Lite	-	Gold	45.81	34.19	0	20.00	42.58	36.77	0	20.65
		RAG+	49.03	43.23	0	7.74	47.10	48.39	0	4.52
		RAG	47.10	45.81	5.16	1.94	39.35	54.84	0.65	5.16
GPT-5-Mini	-	Gold	54.84	26.45	0	18.71	53.55	29.03	0	17.42
		RAG+	62.58	28.39	0	9.03	66.45	29.03	0	4.52
		RAG	60.65	30.32	3.87	5.16	57.42	34.19	5.81	2.58
Gemini-2.5-Flash	-	Gold	47.74	31.61	0	20.65	48.39	34.19	0	17.42
		RAG+	63.23	32.90	0	3.87	64.52	31.61	0	3.87
		RAG	60.00	35.48	3.87	0.65	59.35	37.42	2.58	0.65
Gemini-3-Flash	-	Gold	55.48	25.81	0	18.71	52.90	30.97	0	16.13
		RAG+	72.90	20.00	0	7.10	70.32	23.23	0	6.45
		RAG	67.10	24.52	5.81	2.58	64.52	29.03	3.87	2.58
Claude-Haiku-4.5	-	Gold	49.03	28.39	0	22.58	46.45	30.32	0	23.23
		RAG+	58.06	34.19	0	7.74	53.55	36.77	0	9.68
		RAG	60.00	30.32	6.45	3.23	50.32	38.06	6.45	5.16
Gemini-3.1-Pro	-	Gold	46.45	32.90	0	20.65	47.74	32.90	0	19.35
		RAG+	56.77	35.48	0	7.74	49.68	41.94	0	8.39
		RAG	49.68	38.71	9.03	2.58	51.61	38.71	8.39	1.29
GPT-5.4	-	Gold	49.68	20.65	0	29.68	51.61	19.35	0	29.03
		RAG+	63.87	20.00	0	16.13	61.94	20.00	0	18.06
		RAG	60.00	18.06	14.19	7.74	58.71	18.06	14.84	8.39
Claude-Sonnet-4.6	-	Gold	62.58	23.87	0	13.55	60.00	28.39	0	11.61
		RAG+	77.42	18.06	0	4.52	75.48	19.35	0	5.16
		RAG	76.77	18.71	2.58	1.94	74.19	18.71	3.87	3.23

Table 10: Share of accurate/inaccurate answers and correct/incorrect refusals (%). AccAns/InaccAns denote accurate/inaccurate answers; CorrRef/IncorrRef denote correct/incorrect refusals. Answers with a correctness score of 4–5 from the LLM judge are considered accurate, while scores of 1–3 are considered inaccurate.

model	size	setting	RefRate	F1-macro	RefPr	RefRec	RefF1	NonRefPr	NonRefRec	NonRefF1
Ministral-3B	3B	Gold	17.42	-	-	-	-	100	82.58	90.46
		RAG+	29.68	-	-	-	-	100	70.32	82.58
		RAG	32.26	56.74	38.00	42.22	40.00	75.24	71.82	73.49
Gemma-3-4B-it	4B	Gold	16.13	-	-	-	-	100	83.87	91.23
		RAG+	1.94	-	-	-	-	100	98.06	99.02
		RAG	5.16	41.96	12.50	2.22	3.77	70.07	93.64	80.16
Ministral-8B	8B	Gold	0	-	-	-	-	100	100	100
		RAG+	1.94	-	-	-	-	100	98.06	99.02
		RAG	1.29	43.57	50.00	2.22	4.26	71.24	99.09	82.89
Qwen3.5-9B	9B	Gold	25.16	-	-	-	-	100	74.84	85.61
		RAG+	9.03	-	-	-	-	100	90.97	95.27
		RAG	13.55	59.58	57.14	26.67	36.36	75.37	91.82	82.79
Gemma-3-12B-it	12B	Gold	4.52	-	-	-	-	100	95.48	97.69
		RAG+	21.94	-	-	-	-	100	78.06	87.68
		RAG	21.94	48.19	26.47	20	22.78	70.25	77.27	73.59
Ministral-14B	14B	Gold	1.94	-	-	-	-	100	98.06	99.02
		RAG+	8.39	-	-	-	-	100	91.61	95.62
		RAG	6.45	45.85	30	6.67	10.91	71.03	93.64	80.78
GPT-oss-20B	20B	Gold	22.58	-	-	-	-	100	77.42	87.27
		RAG+	21.94	-	-	-	-	100	78.06	87.68
		RAG	17.42	52.96	37.04	22.22	27.78	72.66	84.55	78.15
Gemma-3-27B-it	27B	Gold	9.68	-	-	-	-	100	90.32	94.92
		RAG+	5.81	-	-	-	-	100	94.19	97.01
		RAG	4.52	46.85	42.86	6.67	11.54	71.62	96.36	82.17
Qwen3.5-27B	27B	Gold	26.45	-	-	-	-	100	73.55	84.76
		RAG+	7.74	-	-	-	-	100	92.26	95.97
		RAG	11.61	61.15	66.67	26.67	38.1	75.91	94.55	84.21
Qwen3-30B-A3B-Instruct	30B	Gold	1.29	-	-	-	-	100	98.71	99.35
		RAG+	3.87	-	-	-	-	100	96.13	98.03
		RAG	1.94	43.30	33.33	2.22	4.17	71.05	98.18	82.44
GLM-4.7-Flash	30B	Gold	3.23	-	-	-	-	100	96.77	98.36
		RAG+	0	-	-	-	-	100	100	100
		RAG	1.94	45.77	66.67	4.44	8.33	71.71	99.09	83.21
Qwen3.5-35B-A3B	35B	Gold	25.16	-	-	-	-	100	74.84	85.61
		RAG+	12.26	-	-	-	-	100	87.74	93.47
		RAG	16.13	63.10	60	33.33	42.86	76.92	90.91	83.33
Llama-3.3-70B-Instruct	70B	Gold	9.03	-	-	-	-	100	90.97	95.27
		RAG+	12.9	-	-	-	-	100	87.1	93.1
		RAG	9.68	50.40	40	13.33	20	72.14	91.82	80.8
Qwen3-Next-80B-A3B-Instruct	80B	Gold	8.39	-	-	-	-	100	91.61	95.62
		RAG+	4.52	-	-	-	-	100	95.48	97.69
		RAG	4.52	53.79	85.71	13.33	23.08	73.65	99.09	84.5
Llama-4-Scout	109B	Gold	10.97	-	-	-	-	100	89.03	94.2
		RAG+	6.45	-	-	-	-	100	93.55	96.67
		RAG	5.81	43.94	22.22	4.44	7.41	70.55	93.64	80.47
GPT-oss-120B	120B	Gold	14.84	-	-	-	-	100	85.16	91.99
		RAG+	9.68	-	-	-	-	100	90.32	94.92
		RAG	10.97	55.65	52.94	20	29.03	73.91	92.73	82.26
Qwen3.5-122B-A10B	122B	Gold	34.84	-	-	-	-	100	65.16	78.91
		RAG+	10.97	-	-	-	-	100	89.03	94.2
		RAG	16.77	62.55	57.69	33.33	42.25	76.74	90	82.85
Qwen3-235B-A22B	235B	Gold	6.45	-	-	-	-	100	93.55	96.67
		RAG+	3.23	-	-	-	-	100	96.77	98.36
		RAG	5.16	48.79	50	8.89	15.09	72.11	96.36	82.49
Qwen3.5-397B-A17B	397B	Gold	25.16	-	-	-	-	100	74.84	85.61
		RAG+	11.61	-	-	-	-	100	88.39	93.84
		RAG	14.19	60.97	59.09	28.89	38.81	75.94	91.82	83.13
Llama-4-Maverick	400B	Gold	10.32	-	-	-	-	100	89.68	94.56
		RAG+	3.87	-	-	-	-	100	96.13	98.03
		RAG	3.23	49.92	80	8.89	16	72.67	99.09	83.85
Mistral-Large-2512	675B	Gold	6.45	-	-	-	-	100	93.55	96.67
		RAG+	0	-	-	-	-	100	100	100
		RAG	0	41.51	0	0	0	70.97	100	83.02
DeepSeek-v3.2	685B	Gold	9.68	-	-	-	-	100	90.32	94.92
		RAG+	4.52	-	-	-	-	100	95.48	97.69
		RAG	3.87	49.54	66.67	8.89	15.69	72.48	98.18	83.4
GLM-5	754B	Gold	23.87	-	-	-	-	100	76.13	86.45
		RAG+	8.39	-	-	-	-	100	91.61	95.62
		RAG	9.03	57.09	64.29	20	30.51	74.47	95.45	83.67
Kimi-K2-Instruct-0905	1T	Gold	4.52	-	-	-	-	100	95.48	97.69
		RAG+	10.32	-	-	-	-	100	89.68	94.56
		RAG	5.16	44.24	25	4.44	7.55	70.75	94.55	80.93
Kimi-K2-Thinking	1T	Gold	33.55	-	-	-	-	100	66.45	79.84
		RAG+	14.84	-	-	-	-	100	85.16	91.99
		RAG	13.55	63.43	66.67	31.11	42.42	76.87	93.64	84.43
Kimi-K2.5	1.1T	Gold	32.9	-	-	-	-	100	67.1	80.31
		RAG+	18.06	-	-	-	-	100	81.94	90.07
		RAG	18.71	68.05	65.52	42.22	51.35	79.37	90.91	84.75
GPT-5-Nano	-	Gold	12.9	-	-	-	-	100	87.1	93.1
		RAG+	10.97	-	-	-	-	100	89.03	94.2
		RAG	12.26	50.77	36.84	15.56	21.88	72.06	89.09	79.67
Gemini-3.1-Flash-Lite	-	Gold	20.65	-	-	-	-	100	79.35	88.49
		RAG+	4.52	-	-	-	-	100	95.48	97.69
		RAG	5.81	57.39	88.89	17.78	29.63	74.66	99.09	85.16
GPT-5-Mini	-	Gold	17.42	-	-	-	-	100	82.58	90.46
		RAG+	4.52	-	-	-	-	100	95.48	97.69
		RAG	8.39	57.58	69.23	20	31.03	74.65	96.36	84.13
Gemini-2.5-Flash	-	Gold	17.42	-	-	-	-	100	82.58	90.46
		RAG+	3.87	-	-	-	-	100	96.13	98.03
		RAG	3.23	49.92	80	8.89	16	72.67	99.09	83.85
Gemini-3-Flash	-	Gold	16.13	-	-	-	-	100	83.87	91.23
		RAG+	6.45	-	-	-	-	100	93.55	96.67
		RAG	6.45	52.48	60	13.33	21.82	73.1	96.36	83.14
Claude-Haiku-4.5	-	Gold	23.23	-	-	-	-	100	76.77	86.86
		RAG+	9.68	-	-	-	-	100	90.32	94.92
		RAG	11.61	57.17	55.56	22.22	31.75	74.45	92.73	82.59
Gemini-3.1-Pro	-	Gold	19.35	-	-	-	-	100	80.65	89.29
		RAG+	8.39	-	-	-	-	100	91.61	95.62
		RAG	11.61	63.15	72.22	28.89	41.27	76.64	95.45	85.02
GPT-5.4	-	Gold	29.03	-	-	-	-	100	70.97	83.02
		RAG+	18.06	-	-	-	-	100	81.94	90.07
		RAG	23.23	70.75	63.89	51.11	56.79	81.51	88.18	84.72
Claude-Sonnet-4.6	-	Gold	11.61	-	-	-	-	100	88.39	93.84
		RAG+	5.16	-	-	-	-	100	94.84	97.35
		RAG	6.45	52.48	60	13.33	21.82	73.1	96.36	83.14

Table 11: Refusal metrics for Dutch. We report the refusal rate (RefRate) and macro-averaged F1 over refusal versus non-refusal, along with class-wise precision, recall, and F1 for refusals (RefPr/RefRec/RefF1) and non-refusals (NonRefPr/NonRefRec/NonRefF1). “-” indicates undefined metrics.

model	size	setting	RefRate	F1-macro	RefPr	RefRec	RefF1	NonRefPr	NonRefRec	NonRefF1
Ministral-3B	3B	Gold	15.48	-	-	-	-	100	84.52	91.61
		RAG+	33.55	-	-	-	-	100	66.45	79.84
		RAG	37.42	51.29	31.03	40	34.95	72.16	63.64	67.63
Gemma-3-4B-it	4B	Gold	8.39	-	-	-	-	100	91.61	95.62
		RAG+	2.58	-	-	-	-	100	97.42	98.69
		RAG	2.58	47.88	75	6.67	12.24	72.19	99.09	83.52
Ministral-8B	8B	Gold	0	-	-	-	-	100	100	100
		RAG+	0	-	-	-	-	100	100	100
		RAG	0	41.51	0	0	0	70.97	100	83.02
Gemma-3-12B-it	12B	Gold	0.65	-	-	-	-	100	99.35	99.68
		RAG+	10.32	-	-	-	-	100	89.68	94.56
		RAG	9.68	44.2	20	6.67	10	70	89.09	78.4
Qwen3.5-9B	9B	Gold	22.58	-	-	-	-	100	77.42	87.27
		RAG+	7.1	-	-	-	-	100	92.9	96.32
		RAG	12.26	58.65	57.89	24.44	34.38	75	92.73	82.93
Ministral-14B	14B	Gold	1.29	-	-	-	-	100	98.71	99.35
		RAG+	3.87	-	-	-	-	100	96.13	98.03
		RAG	3.23	49.92	80	8.89	16	72.67	99.09	83.85
GPT-oss-20B	20B	Gold	18.06	-	-	-	-	100	81.94	90.07
		RAG+	15.48	-	-	-	-	100	84.52	91.61
		RAG	16.13	50.18	32	17.78	22.86	71.54	84.55	77.5
Gemma-3-27B-it	27B	Gold	12.26	-	-	-	-	100	87.74	93.47
		RAG+	7.1	-	-	-	-	100	92.9	96.32
		RAG	6.45	50.27	50	11.11	18.18	72.41	95.45	82.35
Qwen3.5-27B	27B	Gold	22.58	-	-	-	-	100	77.42	87.27
		RAG+	6.45	-	-	-	-	100	93.55	96.67
		RAG	7.74	55.93	66.67	17.78	28.07	74.13	96.36	83.79
Qwen3-30B-A3B-Instruct	30B	Gold	3.23	-	-	-	-	100	96.77	98.36
		RAG+	3.23	-	-	-	-	100	96.77	98.36
		RAG	3.87	47.2	50	6.67	11.76	71.81	97.27	82.63
GLM-4.7-Flash	30B	Gold	0	-	-	-	-	100	100	100
		RAG+	0	-	-	-	-	100	100	100
		RAG	0	41.51	0	0	0	70.97	100	83.02
Qwen3.5-35B-A3B	35B	Gold	27.1	-	-	-	-	100	72.9	84.33
		RAG+	9.68	-	-	-	-	100	90.32	94.92
		RAG	12.9	62.04	65	28.89	40	76.3	93.64	84.08
Llama-3.3-70B-Instruct	70B	Gold	9.03	-	-	-	-	100	90.97	95.27
		RAG+	10.97	-	-	-	-	100	89.03	94.2
		RAG	16.13	57.56	48	26.67	34.29	74.62	88.18	80.83
Qwen3-Next-80B-A3B-Instruct	80B	Gold	9.03	-	-	-	-	100	90.97	95.27
		RAG+	4.52	-	-	-	-	100	95.48	97.69
		RAG	5.16	51.07	62.5	11.11	18.87	72.79	97.27	83.27
Llama-4-Scout	109B	Gold	9.03	-	-	-	-	100	90.97	95.27
		RAG+	18.06	-	-	-	-	100	81.94	90.07
		RAG	13.55	48.03	28.57	13.33	18.18	70.9	86.36	77.87
GPT-oss-120B	120B	Gold	14.19	-	-	-	-	100	85.81	92.36
		RAG+	7.1	-	-	-	-	100	92.9	96.32
		RAG	10.32	52.04	43.75	15.56	22.95	72.66	91.82	81.12
Qwen3.5-122B-A10B	122B	Gold	28.39	-	-	-	-	100	71.61	83.46
		RAG+	8.39	-	-	-	-	100	91.61	95.62
		RAG	9.68	64.87	86.67	28.89	43.33	77.14	98.18	86.4
Qwen3-235B-A22B	235B	Gold	4.52	-	-	-	-	100	95.48	97.69
		RAG+	3.23	-	-	-	-	100	96.77	98.36
		RAG	5.16	48.79	50	8.89	15.09	72.11	96.36	82.49
Qwen3.5-397B-A17B	397B	Gold	27.74	-	-	-	-	100	72.26	83.9
		RAG+	9.68	-	-	-	-	100	90.32	94.92
		RAG	10.97	65.73	82.35	31.11	45.16	77.54	97.27	86.29
Llama-4-Maverick	400B	Gold	10.32	-	-	-	-	100	89.68	94.56
		RAG+	13.55	-	-	-	-	100	86.45	92.73
		RAG	12.26	56.68	52.63	22.22	31.25	74.26	91.82	82.11
Mistral-Large-2512	675B	Gold	3.87	-	-	-	-	100	96.13	98.03
		RAG+	0	-	-	-	-	100	100	100
		RAG	0	41.51	0	0	0	70.97	100	83.02
DeepSeek-v3.2	685B	Gold	23.23	-	-	-	-	100	76.77	86.86
		RAG+	11.61	-	-	-	-	100	88.39	93.84
		RAG	14.19	62.87	63.64	31.11	41.79	76.69	92.73	83.95
GLM-5	754B	Gold	23.87	-	-	-	-	100	76.13	86.45
		RAG+	6.45	-	-	-	-	100	93.55	96.67
		RAG	12.26	64.56	73.68	31.11	43.75	77.21	95.45	85.37
Kimi-K2-Instruct-0905	1T	Gold	7.74	-	-	-	-	100	92.26	95.97
		RAG+	3.23	-	-	-	-	100	96.77	98.36
		RAG	5.81	52.91	66.67	13.33	22.22	73.29	97.27	83.59
Kimi-K2-Thinking	1T	Gold	33.55	-	-	-	-	100	66.45	79.84
		RAG+	11.61	-	-	-	-	100	88.39	93.84
		RAG	13.55	65.35	71.43	33.33	45.45	77.61	94.55	85.25
Kimi-K2.5	1.1T	Gold	29.68	-	-	-	-	100	70.32	82.58
		RAG+	16.13	-	-	-	-	100	83.87	91.23
		RAG	18.06	68.64	67.86	42.22	52.05	79.53	91.82	85.23
GPT-5-Nano	-	Gold	13.55	-	-	-	-	100	86.45	92.73
		RAG+	10.97	-	-	-	-	100	89.03	94.2
		RAG	12.26	48.81	31.58	13.33	18.75	71.32	88.18	78.86
Gemini-3.1-Flash-Lite	-	Gold	20	-	-	-	-	100	80	88.89
		RAG+	7.74	-	-	-	-	100	92.26	95.97
		RAG	7.1	56.41	72.73	17.78	28.57	74.31	97.27	84.25
GPT-5-Mini	-	Gold	18.71	-	-	-	-	100	81.29	89.68
		RAG+	9.03	-	-	-	-	100	90.97	95.27
		RAG	9.03	50.81	42.86	13.33	20.34	72.34	92.73	81.27
Gemini-2.5-Flash	-	Gold	20.65	-	-	-	-	100	79.35	88.49
		RAG+	3.87	-	-	-	-	100	96.13	98.03
		RAG	4.52	53.79	85.71	13.33	23.08	73.65	99.09	84.5
Gemini-3-Flash	-	Gold	18.71	-	-	-	-	100	81.29	89.68
		RAG+	7.1	-	-	-	-	100	92.9	96.32
		RAG	8.39	57.58	69.23	20	31.03	74.65	96.36	84.13
Claude-Haiku-4.5	-	Gold	22.58	-	-	-	-	100	77.42	87.27
		RAG+	7.74	-	-	-	-	100	92.26	95.97
		RAG	9.68	58.67	66.67	22.22	33.33	75	95.45	84
Gemini-3.1-Pro	-	Gold	20.65	-	-	-	-	100	79.35	88.49
		RAG+	7.74	-	-	-	-	100	92.26	95.97
		RAG	11.61	65.14	77.78	31.11	44.44	77.37	96.36	85.83
GPT-5.4	-	Gold	29.68	-	-	-	-	100	70.32	82.58
		RAG+	16.13	-	-	-	-	100	83.87	91.23
		RAG	21.94	70.27	64.71	48.89	55.7	80.99	89.09	84.85
Claude-Sonnet-4.6	-	Gold	13.55	-	-	-	-	100	86.45	92.73
		RAG+	4.52	-	-	-	-	100	95.48	97.69
		RAG	4.52	49.17	57.14	8.89	15.38	72.3	97.27	82.95

Table 12: Refusal metrics for French. We report the refusal rate (RefRate) and macro-averaged F1 over refusal versus non-refusal, along with class-wise precision, recall, and F1 for refusals (RefPr/RefRec/RefF1) and non-refusals (NonRefPr/NonRefRec/NonRefF1). “-” indicates undefined metrics.

Model	Size	Source
TF-IDF	-	Sparck Jones (1972)
BM25	-	Robertson et al. (1994)
word2vec	-	Mikolov et al. (2013b,a); Tulkens et al. (2016); Fauconnier (2015)
fastText	-	Bojanowski et al. (2017); Grave et al. (2018)
static-similarity-mrl-multilingual-v1	-	Reimers and Gurevych (2019)
E5-small-trm-nl	0.04B	Lotfi et al. (2025a)
mE5-small	0.1B	Wang et al. (2024b)
E5-base-trm-nl	0.1B	Lotfi et al. (2025a)
potion-multilingual-128M	0.1B	Tulkens and van Dongen (2024)
mContriever	0.2B	Izacard et al. (2021)
DPR-XM	0.3B	Louis et al. (2025)
mE5-base	0.3B	Wang et al. (2024b)
mGTE	0.3B	Zhang et al. (2024a)
E5-large-trm-nl	0.4B	Lotfi et al. (2025a)
LaBSE	0.5B	Feng et al. (2022)
mE5-large	0.6B	Wang et al. (2024b)
mE5-large-instruct	0.6B	Wang et al. (2024b)
BGE-M3	0.6B	Chen et al. (2024a)
snowflake-arctic-embed-l-v2.0	0.6B	Yu et al. (2024)
jina-embeddings-v3	0.6B	Sturua et al. (2025)
E5-mistral-7b	7B	Wang et al. (2024a, 2022)
BGE-Mult.-Gemma2	9B	Chen et al. (2024a); Xiao et al. (2024)
voyage-2-law	-	VoyageAI (2024)
voyage-3-large	-	VoyageAI (2025)
embedding-3-large	-	OpenAI (2025b)
mmarco-mMiniLMv2-L12-H384-v1	0.1B	Wang et al. (2021)
BGE-reranker-base	0.3B	Xiao et al. (2024)
GTE-multilingual-reranker-base	0.3B	Zhang et al. (2024b)
Jina-reranker-v2-base-multilingual	0.3B	JinaAI (2025)
BGE-reranker-large	0.6B	Xiao et al. (2024)
BGE-reranker-v2-m3	0.6B	Li et al. (2023); Chen et al. (2024b)
Ministral-3B	3B	Liu et al. (2026)
Gemma-3-4B-it	4B	Team et al. (2025a)
Ministral-8B	8B	Liu et al. (2026)
Qwen3.5-9B	9B	QwenTeam (2026)
Gemma-3-12B-it	12B	Team et al. (2025a)
Ministral-14B	14B	Liu et al. (2026)
GPT-oss-20B	20B	OpenAI (2025a)
Gemma-3-27B-it	27B	Team et al. (2025a)
GLM-4.7-Flash	30B	Team et al. (2025b)
Qwen3-30B-A3B-Instruct-2507	30B	Team (2025)
Qwen3.5-35B-A3B	35B	QwenTeam (2026)
Llama-3.3-70B-Instruct	70B	Meta (2025)
Qwen3-Next-80B-A3B-Instruct	80B	Team (2025); Yang et al. (2025)
Llama-4-Scout	109B	Meta (2026)
GPT-oss-120B	120B	OpenAI (2025a)
Qwen3.5-122B-A10B	122B	QwenTeam (2026)
Qwen3-235B-A22B-Instruct-2507	235B	Team (2025); Yang et al. (2025)
Qwen3.5-122B-A10B	122B	QwenTeam (2026)
Qwen3.5-397B-A17B	397B	QwenTeam (2026)
Llama-4-Maverick	400B	Meta (2026)
Mistral-Large-2512	675B	Mistral AI (2025)
DeepSeek-v3.2	685B	DeepSeek-AI (2025)
GLM-5	754B	GLM-5-Team et al. (2026)
Kimi-K2-Instruct-0905	1T	Team et al. (2026)
Kimi-K2-Thinking	1T	Team et al. (2026)
Kimi-K2.5	1.1T	Moonshot AI (2026)
GPT-5-Nano	-	OpenAI (2025)
Gemini-3.1-Flash-Lite	-	DeepMind (2026)
GPT-5-Mini	-	OpenAI (2025)
Gemini-2.5-Flash	-	DeepMind (2025a)
Gemini-3-Flash	-	DeepMind (2025b)
Claude-Haiku-4.5	-	Anthropic (2025)
Gemini-3.1-Pro	-	DeepMind (2025c)
GPT-5.4	-	OpenAI (2025c)
Claude-Sonnet-4.6	-	Anthropic (2026)

Table 13: Models used for RAG, ordered by model size when available. The sections correspond to retrieval, reranking, and generative models, respectively.

Prompt Text
<p>You are a Senior Legal Evaluator and expert attorney. Your job is to evaluate the quality of a response generated by an AI Legal Assistant by comparing it against a verified "Gold Standard" answer.</p> <p>You will be provided with:</p> <ol style="list-style-type: none"> <li>1. A Legal Question.</li> <li>2. A Gold Standard Answer (created by human experts).</li> <li>3. A System Response (generated by the AI).</li> </ol> <p>You must rate the System Response on a scale of 1 to 5 based on how well the legal reasoning and factual content align with the Gold Standard.</p> <ul style="list-style-type: none"> <li>- <b>Content Focus:</b> Focus ONLY on the textual content, legal accuracy, and reasoning.</li> <li>- <b>Strict Adherence:</b> The Gold Standard is the absolute truth. If the System Response contradicts the Gold Standard, it is wrong, even if you believe the Gold Standard might be incomplete.</li> <li>- <b>completeness:</b> The System Response must contain the critical legal elements present in the Gold Standard.</li> </ul> <p>Assess the response below and return a JSON object with two fields:</p> <ul style="list-style-type: none"> <li>- <code>"score"</code>: an integer between 1 and 5.</li> </ul> <ol style="list-style-type: none"> <li>1. <b>Critical Failure / Incorrect</b> The response implies the opposite legal conclusion to the Gold Standard, provides dangerous legal advice, or is completely irrelevant to the question.</li> <li>2. <b>Poor / Significant Omissions</b> The conclusion is vague or partially incorrect. It misses the central legal argument or key fact found in the Gold Standard. It may contain hallucinations.</li> <li>3. <b>Acceptable / Partially Complete</b> The response captures the general legal principle correctly but misses important nuances, exceptions, or specific details present in the Gold Standard. It is legally safe but not comprehensive.</li> <li>4. <b>Good / Mostly Accurate</b> The response aligns with the Gold Standard in conclusion and reasoning. It may miss very minor details that do not alter the legal outcome.</li> <li>5. <b>Excellent / Semantically Equivalent</b> The response is logically and factually equivalent to the Gold Standard. It captures all key legal elements, reasoning, and conclusions. (Difference in wording or structure is acceptable).</li> </ol> <ul style="list-style-type: none"> <li>- <code>"reason"</code>: a brief explanation for why the score was given. This must mention specific strengths or shortcomings, referencing relevant details from the input. Do <b>not</b> quote the score itself in the explanation.</li> </ul> <p>Your explanation should:</p> <ul style="list-style-type: none"> <li>- <code>{reasoning_expectation}</code></li> <li>- Mention key details from the test case parameters.</li> <li>- Be concise, clear, and focused on the evaluation logic.</li> </ul> <p>Only return valid JSON. Do <b>not</b> include any extra commentary or text.</p> <p>—</p> <p>Test Case: {test_case_content} Parameters: {parameters}</p> <p>—</p> <p><b>Example JSON:</b></p> <pre>{   "reason": "your concise and informative reason here",   "score": 1 }</pre> <p>JSON: ""</p>

Table 14: DeepEval G-Eval prompt for answer correctness.

Language	Prompt
French	Étant donné une question juridique, récupère les documents qui peuvent aider à y répondre
Dutch	Gegeven een juridische vraag, haal documenten op die kunnen helpen bij het beantwoorden van de vraag

Table 15: Prompts used for E5-large-instruct, BGE-Gemma2 and E5-mistral for the retrieval task.

<p><b>System Prompt</b></p> <p>You are an expert legal assistant specializing in Belgian law.  Your task is to answer legal questions to the best of your knowledge of Belgian law.  You respond exclusively in \$answer_language and prioritize legal accuracy.</p>
<p><b>User Prompt</b></p> <p>Instructions:</p> <ol style="list-style-type: none"> <li>The legal context is provided as a JSON array of articles. Each article has the following structure: <ul style="list-style-type: none"> <li>- "id": the unique identifier of the article</li> <li>- "text": the text of the article excerpt</li> </ul> </li> <li>Carefully analyze all articles in the legal context and assess their relevance to the legal question.</li> <li>Answer the question <b>ONLY IF</b> the context is sufficient: <ul style="list-style-type: none"> <li>- The context must contain all necessary rules or conditions to answer the question.</li> <li>- If any essential condition is missing, unclear, or cannot be derived from the provided texts, do not answer.</li> <li>- If relevant articles conflict on a key condition and the conflict cannot be resolved using only the context, do not answer.</li> <li>- Use <b>ONLY</b> the "text" fields. Do not rely on external knowledge or assumptions.</li> </ul> </li> <li>Output format requirements: <ul style="list-style-type: none"> <li>- Return a JSON array of objects: <pre>["text": "...", "supported_sources": ["id1", "id2"], ...]</pre> </li> <li>- Each object represents exactly one answer paragraph.</li> <li>- <b>EVERY</b> paragraph must be directly supported by one or more article IDs.</li> <li>- "supported_sources" must be a valid JSON array of strings (double quotes).</li> <li>- Include <b>ONLY</b> article IDs that appear in the provided legal context.</li> <li>- Include <b>ONLY</b> IDs that directly support the corresponding paragraph text.</li> <li>- Do not include irrelevant or speculative citations.</li> </ul> </li> <li>If the context is insufficient, incomplete, contradictory, or irrelevant, return exactly: <pre>["text": "Insufficient context", "supported_sources": []]</pre> </li> </ol> <p>Legal question: \$question  Regions involved: \$regions  Topics: \$topics  Legal context (article excerpts):  \$context</p> <p>Output the JSON array immediately. Do not include any preamble.</p>

Table 16: System and user prompts used by the LLMs in the RAG experiments.

# VLA-Forget: Vision-Language-Action Unlearning for Embodied Foundation Models

Ravi Ranjan\*

Florida International University  
Miami, USA  
rkuma031@fiu.edu

Agoritsa Polyzou

Florida International University  
Miami, USA  
apolyzou@fiu.edu

## Abstract

Vision-language-action (VLA) models are emerging as embodied foundation models for robotic manipulation, but their deployment introduces a new unlearning challenge: removing unsafe, spurious, or privacy-sensitive behaviors without degrading perception, language grounding, and action control. In OpenVLA-style policies, behavior is produced through a fused visual encoder, a cross-modal projector, and a language backbone that predicts tokenized robot actions, so undesirable knowledge can be distributed across perception, alignment, and reasoning/action layers rather than confined to a single module. Consequently, partial unlearning applied only to the vision stack or only to the language backbone is often insufficient, while conventional unlearning baselines designed for standalone vision or language models may leave residual forgetting or incur unnecessary utility loss in embodied settings. We propose **VLA-Forget**, a hybrid unlearning framework that combines ratio-aware selective editing for perception and cross-modal specificity with layer-selective reasoning/action unlearning for utility-preserving forgetting. VLA-Forget jointly optimizes three objectives: targeted forgetting, perceptual preservation, and reasoning retention, through staged updates over the visual encoder, projector, and upper action-generating transformer blocks. Across forget-set behavior probes and retain-task evaluations, VLA-Forget improves forgetting efficacy by 10%, preserves perceptual specificity by 22%, retains reasoning and task success by 9%, and reduces post-quantization recovery by 55% relative to strong unlearning baselines.

## 1 Introduction

Vision-Language-Action (VLA) models are emerging as a practical route to *embodied foundation models*: policies that translate natural-language instructions and raw visual observations directly

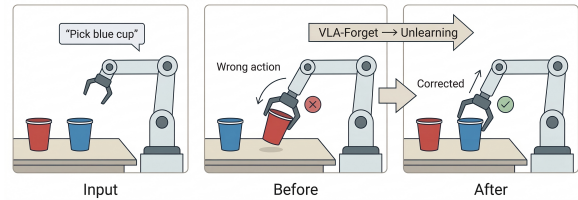


Figure 1: **Motivating failure case for VLA-Forget.** Given the instruction “Pick blue cup,” the policy produces a visually plausible but semantically incorrect action by grasping the red cup. After targeted unlearning, the policy suppresses this spurious instruction-to-action association and recovers instruction-consistent behavior.

into robot actions by leveraging large pretrained vision and language backbones. OpenVLA exemplifies this paradigm as an open 7B VLA trained on 970K real-world robot demonstrations from Open X-Embodiment, combining a fused (DINOv2+SigLIP) visual encoder with a Llama 2 backbone and an action-token prediction interface that enables scalable fine-tuning and deployment on commodity hardware (Kim et al., 2024; O’Neill et al., 2024; Pang et al., 2025).

However, *deployment* of VLA policies introduces an urgent governance and reliability problem that is not fully addressed by today’s training and evaluation pipelines: once a robot policy has been trained, it may retain (i) unsafe or undesirable behaviors from a subset of demonstrations, (ii) privacy or IP-sensitive content embedded in multimodal data, or (iii) spurious shortcuts that appear correct under standard benchmarks but fail under structured distribution shift. The consequences of such failures are amplified in robotics because errors translate into physical actions. Recent evidence further highlights a VLA-specific grounding failure, sometimes described as *linguistic blindness* where policies execute visually plausible trajectories even when the instruction is semantically contradictory, motivating stronger mechanisms for selective behavior re-

\*Corresponding author.

moval and reliable grounding diagnostics (Zhuang et al., 2026).

Figure 1 illustrates the core VLA-specific challenge studied in this work: undesirable behavior may arise from misalignment across perception, cross-modal grounding, or action priors, causing the policy to execute a physically plausible yet instruction-inconsistent action. This motivates *selective unlearning* that removes a targeted erroneous behavior slice while preserving normal scene understanding and non-target task execution. Motivation is further supported by recent evidence that VLA policies remain brittle under visual corruption and grounding perturbations, underscoring the need for post hoc correction mechanisms beyond standard fine-tuning (Lin et al., 2025; Orjuela et al., 2026).

Machine unlearning provides a practical way to remove targeted training influence without full re-training, but VLA policies are harder to unlearn than standard vision or language models. Unlike static predictors, a VLA model is a *closed-loop control policy*, so failure must be judged through embodied behavior rather than output accuracy alone. In OpenVLA, continuous robot actions are represented as *discrete action tokens*, which means unwanted behavior can be encoded jointly in visual features, cross-modal alignment, and instruction-conditioned action priors in the language backbone (Kim et al., 2024).

As a result, unlearning only the vision stack or only the language model is often insufficient: removing a visual trigger may leave the downstream action prior intact, while editing language priors may preserve harmful perceptual shortcuts or incorrect visual-language bindings. Existing baselines were largely designed for unimodal settings and transfer only partially to VLA policies. For example, exact approaches such as SISA require training-time changes (Bourtole et al., 2021), while approximate methods such as SCRUB improve forget-retain trade-offs but do not directly address multimodal component entanglement or control-oriented evaluation (Kurmanji et al., 2023). Moreover, VLA unlearning involves competing retain, forget, and mismatch objectives over large backbones, motivating gradient-conflict mitigation such as PCGrad (Yu et al., 2020).

In this work, we introduce **VLA-Forget**, a hybrid unlearning framework for Vision-Language-Action models that is explicitly *component-aware* and *deployment-oriented*. VLA-Forget (i) targets

perception and cross-modal specificity through ratio-aware selective editing of the visual encoder and projector, (ii) targets reasoning/action utility-preserving forgetting through significance-based selective editing of action-relevant transformer blocks, and (iii) performs staged, adapter-first updates (e.g., LoRA) to enable efficient unlearning with rollback and canary deployment compatibility (Hu et al., 2022; Kim et al., 2025; Liu et al., 2025a). We evaluate unlearning with robotics-centric metrics and unlearning-centric audits (including safety violation risk); furthermore, we evaluate the model’s robustness post-quantization, aligning the evaluation with the realities of VLA deployment (Carlini et al., 2022; Zang et al., 2025; Zhang et al., 2024b) While VLA-Forget improves targeted behavior suppression in benchmarked embodied settings, it is an approximate unlearning method and should not be interpreted as providing certified erasure.

**Key Contributions.** (i) We formalize *VLA unlearning* as a three-goal problem **targeted forgetting** (efficacy), **perceptual preservation** (specificity), and **reasoning retention** (utility) in the presence of an action-token interface and multimodal component entanglement. (ii) We propose **VLA-Forget**, a novel hybrid unlearning pipeline that exploits ratio-aware selective editing for perception/cross-modal specificity with significance-based selective editing for reasoning/action utility preservation, implemented in an adapter-first manner compatible with OpenVLA fine-tuning workflows. (iii) We outline an **evaluation protocol** that couples embodied performance (task success and control stability) with unlearning audits (forget/retain scores and safety violation risk), and incorporates structured contradiction probes to diagnose and prevent spurious “fake success” under OOD instructions.

## 2 Related Work

Vision-language-action (VLA) models extend multimodal foundation models from perception and generation to embodied control. Early systems such as VIMA and RT-2 showed that robot behavior can be conditioned on interleaved visual-language prompts and represented through tokenized actions, while Open X-Embodiment and OpenVLA scaled this paradigm to diverse real-robot data and open 7B policies. Despite this progress, prior VLA research has focused mainly on scaling, transfer, and adaptation, rather than post hoc removal of un-

safe behaviors, undesirable concepts, or sensitive instruction-action associations. (Jiang et al., 2023; Zitkovich et al., 2023; O’Neill et al., 2024; Grover et al., 2026; Kim et al., 2024)

Machine unlearning has progressed from exact retraining-based deletion to approximate updates that balance erasure quality, efficiency, and retained utility. In language models, recent work spans gradient-ascent, retain-regularized, preference-based, and activation-space methods, while emphasizing joint evaluation of forgetting, utility preservation, and privacy leakage. Two recurring issues are especially relevant for embodied policies: broad model-wide edits often cause collateral degradation, and apparently successful forgetting can fail under deployment transformations such as quantization. These limitations motivate selective, structure-aware unlearning over indiscriminate full-model updates. (Bourtole et al., 2021; Maini et al., 2024; Yao et al., 2024; Liu et al., 2025b; Zhang et al., 2024b)

Related work in vision, vision-language, and diffusion models studies forgetting of classes, identities, and concepts in multimodal transformers. SSD and SalUn localize updates through parameter importance or saliency, LoTUS improves scalable approximate unlearning with uncertainty-aware smoothing, and SLUG shows that targeted single-layer editing can sometimes suffice. In generative settings, ESD and UCE edit text-conditioned diffusion behavior at the concept level. Collectively, these methods expose a central trade-off: aggressive updates improve forgetting but may damage retention. (Foster et al., 2024; Fan et al., 2023; Spartalis et al., 2025; Cai et al., 2024; Gandikota et al., 2023, 2024)

VLA unlearning differs from prior LLM- or VLM-only settings because undesired behavior may be encoded jointly in visual features, cross-modal bindings, and action-token priors, and must be evaluated through embodied execution rather than text or image outputs alone. Accordingly, VLA-FORGET adopts a component-aware formulation: ratio-aware edits target perception and projector modules, while significance-based selective updates address higher-level reasoning and action-generation components. This yields a hybrid unlearning strategy tailored to embodied foundation models rather than a direct reuse of unimodal forgetting methods. (Zitkovich et al., 2023; Kim et al., 2024; Fan et al., 2023; Cai et al., 2024; Yao et al., 2024)

### 3 Methodology

**Overview.** We propose **VLA-Forget**, a novel unlearning framework for vision-language-action (VLA) policies that removes targeted behaviors while preserving perceptual grounding and action reasoning. We instantiate the method on OpenVLA-style policies, where an input image and language instruction are processed by a fused visual encoder, projected into the language-model embedding space, and decoded into discretized robot action tokens. In OpenVLA, the visual stack combines DINOv2 and SigLIP, the backbone is Llama 2, and the policy predicts normalized 7-DoF actions via tokenized bins, making forgetting inherently distributed across perception, cross-modal alignment, and action generation rather than localized to a single module (Kim et al., 2024).

**Architecture and unlearning target.** Let a VLA policy be

$$f_{\theta}(o, s) = \text{Dec}_{\theta_L} \left( \text{Proj}_{\theta_P} \left( \text{Enc}_{\theta_V}(o) \right), s \right), \quad (1)$$

where  $o$  is the observation image,  $s$  is the instruction,  $\theta_V$  denotes the visual encoder,  $\theta_P$  the MLP projector, and  $\theta_L$  the language/action backbone. The decoder produces an action-token sequence  $y$ , which is de-tokenized into a continuous action  $a \in \mathbb{R}^7$ . We consider an unlearning request  $U$  specifying a target concept or behavior slice, such as a sensitive object, unsafe affordance, or erroneous instruction-to-action mapping. We form three datasets: a forget set  $D_f$ , a retain set  $D_r$ , and a boundary set  $D_m$  containing near-neighbor scenes or instructions that must remain intact.

**Why “VLA” unlearning is necessary.** In VLA models, undesired behavior can originate from three coupled sources: (i) perceptual memorization in  $\theta_V$ , (ii) erroneous visual-to-language binding in  $\theta_P$ , and (iii) action priors or instruction-conditioned hallucinations in  $\theta_L$ . Editing only the vision stack may suppress the visual trigger while leaving the downstream action prior unchanged; editing only the language backbone may preserve a harmful perceptual shortcut. VLA-Forget therefore combines ratio-aware editing for perception and cross-modal specificity with layer-selective unlearning for reasoning and action retention.

#### 3.1 Hybrid Selective Localization

**Perception and projector selection.** For each candidate visual or projector layer  $l$ , we compute

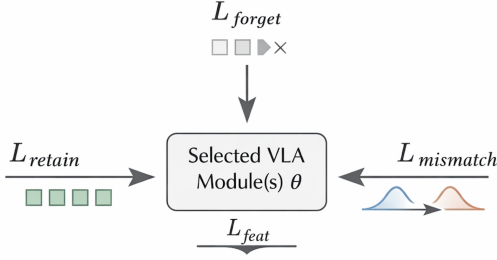


Figure 2: **Illustration of the unlearning objectives on a selected module.** Eq. 6 suppresses the targeted behavior, Eq. 5 preserves non-targeted behavior, and Eq. 7 discourages shallow forgetting by pushing the updated policy away from the original forgotten response.

forget and retain gradients as follows:

$$g_l^f = \nabla_{\theta_l} \mathcal{L}_{\text{forget}}, \quad \text{and} \quad g_l^r = \nabla_{\theta_l} \mathcal{L}_{\text{retain}}. \quad (2)$$

We then assign a ratio-aware score

$$\phi(l) = \frac{\|g_l^f\|_2}{\|\theta_l\|_2 + \varepsilon} (1 - \cos(g_l^f, g_l^r))^\alpha, \quad (3)$$

and select the top visual/projector layers  $K_V, K_P$  with highest  $\phi(l)$ . This favors parameters that strongly affect forgetting while minimally interfering with retained perception (Yu et al., 2020).

**Reasoning/action layer selection.** For the upper transformer blocks in the language-action backbone, we calculate a significance ratio:

$$\text{Sig}(l) = \frac{\|\nabla_{\theta_l} \mathcal{L}_{\text{forget}}\|_2}{\|\nabla_{\theta_l} \mathcal{L}_{\text{retain}}\|_2 + \varepsilon}. \quad (4)$$

We initialize an editable set  $S_L$  with the top- $k$  layers under  $\text{Sig}(l)$ , and expand it iteratively only if forgetting criteria are unmet. This yields a minimal update set for action-relevant reasoning while avoiding unnecessary global drift (Ranjan et al., 2026; Kurmanji et al., 2023; Zhang et al., 2024b).

### 3.2 Unlearning Objectives

We optimize three complementary objectives aligned with the goals of efficacy, specificity, and utility.

**Retain loss.** To preserve non-targeted behavior, we minimize

$$\begin{aligned} \mathcal{L}_{\text{retain}} = & \mathbb{E}_{(x,y) \sim D_r} [\text{CE}(p_\theta(\cdot | x), y)] \\ & + \beta \mathbb{E}_{x \sim D_r} [\text{KL}(p_{\theta_0}(\cdot | x) \| p_\theta(\cdot | x))]. \end{aligned} \quad (5)$$

where  $\theta_0$  denotes the original model and  $x = (o, s)$ . The first term preserves action-token prediction on retained data, while the KL anchor constrains the updated policy to remain close to the

base model on benign trajectories (Kim et al., 2024; Kurmanji et al., 2023).

**Forget loss.** To suppress the targeted behavior, we maximize the forget-set prediction error:

$$\mathcal{L}_{\text{forget}} = \mathbb{E}_{(x,y) \sim D_f} [\text{CE}(p_\theta(\cdot | x), y)]. \quad (6)$$

Operationally, this is implemented as gradient ascent on  $\mathcal{L}_{\text{forget}}$  (Jin et al., 2025).

**Mismatch loss.** To avoid shallow forgetting and reduce recovery to the pre-unlearning behavior, we maximize distributional divergence on forgotten samples:

$$\mathcal{L}_{\text{mismatch}} = \mathbb{E}_{x \sim D_f} [\text{KL}(p_\theta(\cdot | x) \| p_{\theta_0}(\cdot | x))]. \quad (7)$$

As shown in Fig. 2, the three objectives act together on the selected module to suppress the target behavior, preserve retained behavior, and reduce shallow recovery to the pre-unlearning policy. (Appendix A.1)

**Perceptual preservation loss.** To preserve non-targeted visual grounding, we distill internal features on  $D_r \cup D_m$ :

$$\begin{aligned} \mathcal{L}_{\text{feat}} = & \mathbb{E}_{x \sim D_r \cup D_m} [\|h_\theta^V(x) - h_{\theta_0}^V(x)\|_2^2 \\ & + \gamma \|h_\theta^P(x) - h_{\theta_0}^P(x)\|_2^2], \end{aligned} \quad (8)$$

where  $h^V$  and  $h^P$  are late visual and projector representations.

**Unified objective.** The final optimization is

$$\min_{\theta} \mathcal{L}_{\text{retain}} + \lambda_{\text{feat}} \mathcal{L}_{\text{feat}} - \lambda_f \mathcal{L}_{\text{forget}} - \lambda_m \mathcal{L}_{\text{mismatch}}. \quad (9)$$

### 3.3 Training and Unlearning Procedure

**Adapter-first updates.** We implement VLA-Forget using parameter-efficient adapters over the selected layers while freezing the remaining weights. This design is compatible with the OpenVLA training stack, which supports LoRA-based fine-tuning with `target_modules=all-linear`; thus unlearning can be applied without full-model retraining and can be merged or rolled back at deployment time (Kim et al., 2025; Hu et al., 2022).

**Stage 1: perception unlearning.** We first update LoRA parameters on  $K_V$  to weaken targeted visual evidence while preserving general scene understanding through  $\mathcal{L}_{\text{retain}}$  and  $\mathcal{L}_{\text{feat}}$ . This stage removes object- or scene-level triggers with minimal disruption to unrelated perception.

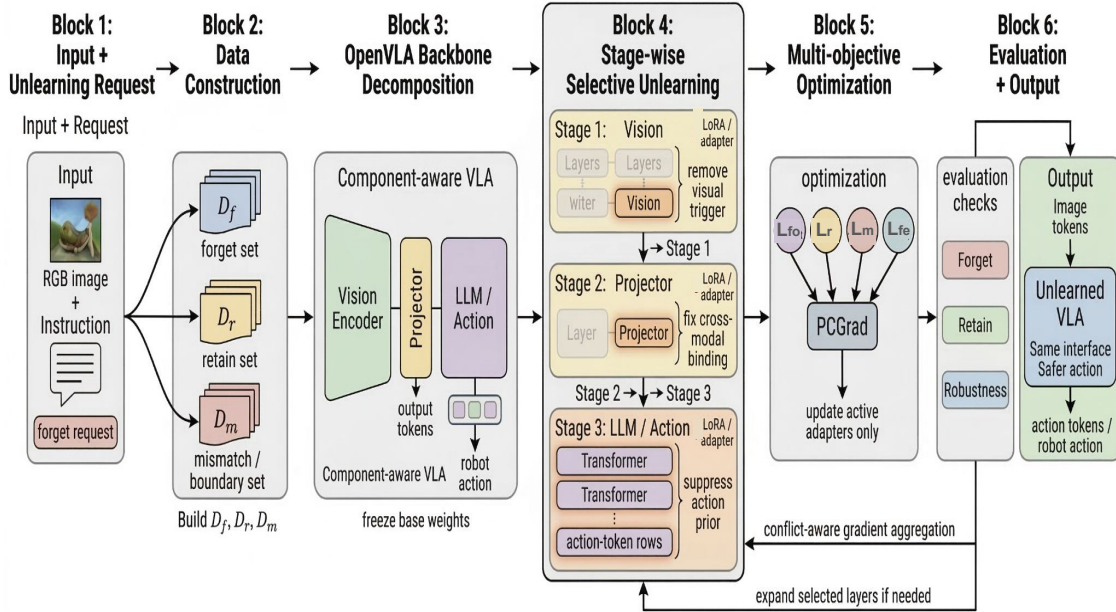


Figure 3: Overview of VLA-Forget. Given an unlearning request, we construct forget, retain, and mismatch sets, then perform staged adapter-based selective updates over the vision encoder, projector, and LLM/action layers with multi-objective PCGrad stabilization, yielding an unlearned VLA policy with preserved interface and retained task utility.

**Stage 2: cross-modal unlearning.** Next, we update  $K_P$  in the projector to break the specific visual-to-language associations responsible for the unwanted behavior. This is essential when the failure is not purely visual, but rather caused by erroneous alignment between image regions and action-relevant tokens.

**Stage 3: reasoning/action unlearning.** Finally, we update the selected upper backbone layers  $S_L$ , and optionally the action-token embedding rows most implicated in the target behavior. This stage suppresses residual instruction-conditioned action priors while maintaining overall task competence.

**Gradient stabilization and stopping.** Because the retain and forget objectives can conflict, we aggregate gradients using projected conflict resolution:

$$g = \text{PCGrad} \left( \nabla \mathcal{L}_{\text{retain}}, -\lambda_f \nabla \mathcal{L}_{\text{forget}}, -\lambda_m \nabla \mathcal{L}_{\text{mismatch}} \right), \quad (10)$$

and apply optimizer steps only to the active adapter parameters. After each round, we evaluate forget efficacy on  $D_f$  and retain utility on  $D_r$ . If forgetting remains insufficient, we expand  $S_L$  by the next highest-significance layer and continue; otherwise, training stops early (Yu et al., 2020).

Figure 3 summarizes the VLA-Forget pipeline as a staged, component-aware unlearning framework. Starting from an unlearning request, the method

constructs forget, retain, and mismatch sets, decomposes the OpenVLA policy into vision, projector, and LLM/action modules, and then applies selective adapter-based updates in three stages: visual trigger removal, cross-modal binding correction, and reasoning/action prior suppression. The optimization block combines retain, forget, mismatch, and perceptual preservation objectives with PCGrad stabilization, while the final evaluation block verifies forgetting efficacy, retained utility, and deployment robustness before producing the unlearned VLA policy.

**Interaction with VLA policies.** VLA-Forget is designed to preserve the native VLA interface: image and instruction inputs remain unchanged, and the model still predicts action tokens followed by the standard de-tokenization and action unnormalization pipeline used by OpenVLA. This makes the method directly deployable in existing VLA control stacks while enabling targeted post hoc removal of unsafe, private, or spurious behaviors (Zang et al., 2025; Pang et al., 2025).

The complete unlearning pipeline is detailed in Algorithm 1, with its associated supporting procedures provided in Algorithm 2; corresponding pseudo-code is available in Appendix A.

Table 1: Unlearning result on **OpenVLA-7B**. Left: Open X-Embodiment. Right: lerobot/pusht\_image. Higher is better for FC, RC, FAD, and TSR; lower is better for RAD and SVR. Mean  $\pm$  standard deviation over 5 random seeds, with typical variation in the range of  $\pm(1-3)$  points for rate-based metrics (FC, RC, TSR, and SVR) and  $\pm(0.01-0.03)$  for distance-based metrics (FAD and RAD). Best results are shown in **bold** and second-best are underlined.

Method	Open X-Embodiment (OpenVLA-7B)						lerobot/pusht_image (OpenVLA-7B)					
	FC $\uparrow$	RC $\uparrow$	FAD $\uparrow$	RAD $\downarrow$	TSR $\uparrow$	SVR $\downarrow$	FC $\uparrow$	RC $\uparrow$	FAD $\uparrow$	RAD $\downarrow$	TSR $\uparrow$	SVR $\downarrow$
SSD	78	83	0.70	0.28	68	17	82	86	0.73	0.20	55	15
SalUn	89	88	0.76	0.26	71	12	89	88	0.78	0.18	60	11
GA	<b>93</b>	60	<b>0.89</b>	0.45	40	<b>5</b>	<u>94</u>	50	<b>0.91</b>	0.50	22	<b>3</b>
NPO	<u>90</u>	<u>88</u>	0.83	<u>0.23</u>	<u>74</u>	<u>8</u>	92	<u>90</u>	0.85	<u>0.15</u>	<u>65</u>	7
<b>VLA-Forget</b>	<b>93</b>	<b>91</b>	<u>0.88</u>	<b>0.21</b>	<b>78</b>	<b>5</b>	<b>95</b>	<b>94</b>	<u>0.90</u>	<b>0.13</b>	<b>69</b>	<u>4</u>

## 4 Experiments

### 4.1 Experimental Setup

**Models.** We instantiate VLA-FORGET on OpenVLA-7B, a 7B vision-language-action policy trained on large-scale Open X-Embodiment robot data (Kim et al., 2024; O’Neill et al., 2024). The second model is pi0fast-base that predicts continuous robot actions via auto-regressive next-token prediction (Pertsch et al., 2025). Following the VLA setting in our method, we construct three splits for each experiment: a forget set  $D_f$  containing target behaviors to be removed, a retain set  $D_r$  containing non-target behaviors whose utility should be preserved, and a boundary set  $D_m$  containing near-neighbor samples used to reduce shallow or entangled forgetting. Unless otherwise stated, all methods start from the same pretrained checkpoint, use the same forget/retain protocol, and are trained under the same optimization budget.

**Datasets.** We use two complementary benchmarks. First, we build a real-robot benchmark from subsets of Open X-Embodiment (OXE), downloaded via `gsutil`, where each trajectory is paired with a language instruction and organized into forget/retain slices at the trajectory level (O’Neill et al., 2024). Second, we use lerobot/pusht\_image as a controlled synthetic benchmark, where we inject a static text instruction into each instance and form matched forget/retain partitions (Cadene et al., 2026). In the released OpenVLA, we use up to 512 prompted PushT instances with a 30% forget fraction; in the lightweight ablation pipeline, we scale this to up to 4,000 instances and use a 70/15/15 train/validation/test split, reporting forget and retain test performance after unlearning. This combination gives both a realistic OXE setting and a reproducible low-cost benchmark for

rapid ablations.

**Baselines.** We compare against four representative approximate unlearning baselines. **SSD** (Foster et al., 2024) is a retraining-free parameter-dampening method that suppresses weights estimated to be disproportionately important to the forget data. **SalUn** (Fan et al., 2023) is a saliency-based baseline that updates only high-importance weights, making it a strong selective vision-side unlearning comparator. **GA** (Yao and Xu, 2024) performs direct gradient ascent on the forget loss and serves as the standard language-side unlearning baseline. **NPO** (Zhang et al., 2024a) is a preference-based alternative designed to improve forget-retain trade-offs. These baselines span both vision-oriented and language-oriented unlearning regimes, making them suitable comparators for VLA policies.

**Metrics.** We report six metrics, **Forget action loss / cross-entropy (FC)** measures forgetting efficacy on  $D_f$ ; *higher* FC is better because the model should become less able to reproduce the forgotten action mapping. **Retain utility score (RC)**, derived from retain-set cross-entropy, measures preserved utility on  $D_r$ ; *higher* is better. **Forget Accuracy Drop (FAD)** measures the reduction in action-token or exact-match accuracy on the forget split relative to the base model; *higher* is better. **Retain Accuracy Drop (RAD)** measures the corresponding degradation on retained behaviors; *lower* is better. **Task Success Rate (TSR)** evaluates closed-loop policy execution on benchmark tasks; *higher* is better. **Safety Violation Rate (SVR)** measures the frequency of unsafe or disallowed behaviors under target prompts or contradiction probes; *lower* is better. We follow standard VLA and robot manipulation evaluation protocols (Kim et al., 2024; Liu et al., 2023). Together, these metrics capture

Table 2: Unlearning results on lerobot/pi0fast-base under the Open X-Embodiment protocol. Mean  $\pm$  standard deviation over 5 random seeds, with typical variation of  $\pm(1-3)$  points for FC, RC, TSR, and SVR, and  $\pm(0.01-0.04)$  for FAD and RAD. Best results are shown in **bold** and second-best are underlined.

Open X-Embodiment (Pi0-FAST-Base)						
Method	FC $\uparrow$	RC $\uparrow$	FAD $\uparrow$	RAD $\downarrow$	TSR $\uparrow$	SVR $\downarrow$
SSD	76	81	0.68	0.30	65	18
SalUn	87	85	0.74	0.27	68	13
GA	<u>93</u>	57	<b>0.89</b>	0.47	38	<b>6</b>
NPO	89	87	0.82	<u>0.24</u>	<u>72</u>	<u>9</u>
<b>VLA-Forget</b>	<b>94</b>	<b>89</b>	<u>0.88</u>	<b>0.22</b>	<b>75</b>	<b>6</b>

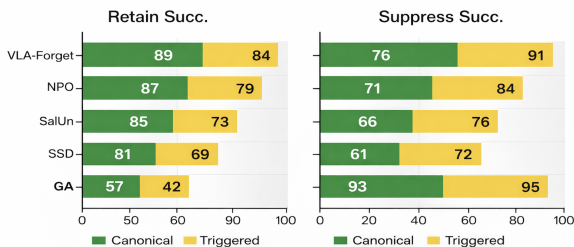


Figure 4: Instruction-conditioned action behavior on pi0fast-base and Open X-Embodiments.

forgetting strength, retained utility, embodied execution quality, and safety, which are all necessary for VLA unlearning.

Detailed experimental setup, hyper-parameter choices, and reproducibility details are provided in Appendix B. Metric definitions and their mathematical formulations are provided in Appendix B.2.

## 4.2 Experiment Results

**Result on OpenVLA.** Table 1 highlights three key observations. VLA-Forget achieves the best overall balance between forgetting and retention across both benchmarks: although GA attains the strongest raw forgetting scores (FC/FAD), it severely degrades retain performance and task success, whereas VLA-Forget preserves the highest RC, the lowest RAD, and the best TSR, indicating substantially better utility preservation. The same trend is consistent on both Open X-Embodiment and PushT, suggesting that the method is not tied to a single dataset but generalizes across broader settings. Lower SVR of VLA-Forget relative to SSD, SalUn, and NPO shows more reliable suppression of undesired behaviors without inducing large action drift, which implies that its component-aware multimodal unlearning is better aligned with safe deployment than purely aggressive forgetting baselines. The safety violation plot of figure 5 shows

Table 3: Quantization robustness after unlearning on OpenVLA-7B with Open X-Embodiment dataset. Results report forget-side metrics under post-training 8-bit and 4-bit quantization.

Method	8-bit		4-bit	
	FC $\uparrow$	SVR $\downarrow$	FC $\uparrow$	SVR $\downarrow$
SSD	76	19	72	23
SalUn	87	8	78	10
GA	82	12	80	16
NPO	85	10	82	13
<b>VLA-Forget</b>	<b>91</b>	<b>6</b>	<b>88</b>	<b>8</b>

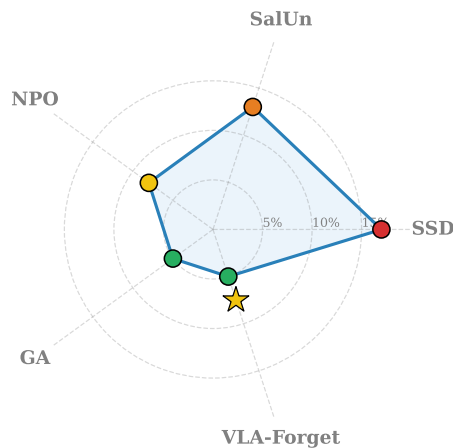


Figure 5: Safety violation rate (SVR) on Open X-Embodiment for different unlearning methods on OpenVLA-7B, where lower values indicate safer post-unlearning behavior.

that VLA-Forget along with GA achieves one of the lowest safety violation rates while maintaining strong overall unlearning performance, indicating a better balance between removing unsafe behaviors and preserving stable policy execution.

**Result on  $\pi 0$  fast.** Table 2 shows that VLA-Forget preserves the same overall pattern as in the OpenVLA results: although GA attains very strong forgetting, it suffers from severe retain-side degradation, whereas VLA-Forget achieves the best overall trade-off by jointly maintaining the highest RC and TSR, the lowest RAD, and near-best forgetting performance. In Fig. 4, Canonical denotes standard retain-task instructions, while Triggered denotes target-trigger or contradictory instructions used to probe the unlearning scope. Higher Retain Succ. indicates better preserved non-target behavior, and higher Suppress Succ. indicates more effective suppression of the unwanted action. VLA-Forget achieves the best overall balance by preserving the strongest retain-task performance under

Table 4: Ablation results of VLA-Forget on **OpenVLA-7B** on Open X-Embodiment dataset, across different component selections.

VLA-Forget applied to	FC $\uparrow$	RC $\uparrow$	FAD $\uparrow$	RAD $\downarrow$	TSR $\uparrow$	SVR $\downarrow$
Vision encoder only	85	82	0.80	0.27	65	12
Projector only	82	89	0.75	0.22	75	15
Language backbone only	90	88	0.85	0.23	74	8
Vision + Projector	88	86	0.84	0.24	72	10
Projector + Language	92	90	0.86	0.22	77	6
Vision + Language	93	87	0.89	0.25	70	6
Full (Vision + Proj + Lang)	93	91	0.88	0.21	78	5
Full, w/o retain-preservation	95	80	0.93	0.30	64	4

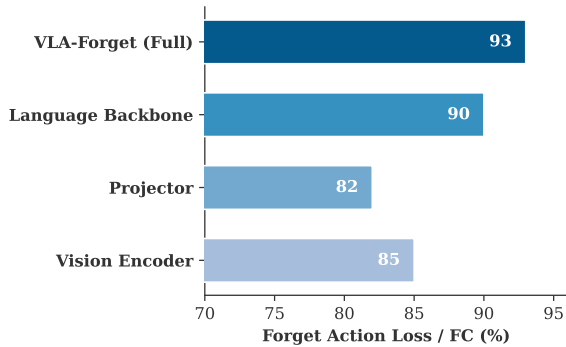


Figure 6: Forget-action comparison across component-level VLA-Forget ablations on OpenVLA-7B and Open X-Embodiment.

canonical instructions while also maintaining high suppression under triggered instructions. *Canonical* corresponds to normal non-target instructions, and *Triggered* corresponds to target-triggered or contradiction-style instructions; in both panels, higher values are better.

**Robustness Result.** Table 3 demonstrates that quantization degrades forgetting quality for all methods, but the drop is largest for broad full-model updates such as GA and NPO, whereas VLA-Forget remains comparatively stable under both 8-bit and 4-bit settings, suggesting stronger robustness to quantization-induced recovery.

### 4.3 Ablation Study

Table 4 shows that unlearning only a single component is insufficient: the language backbone gives stronger forgetting than vision-only or projector-only updates, but the *full* multimodal setting achieves the best overall trade-off, indicating that undesirable action behavior is distributed across perception, alignment, and language-action reasoning modules. The vision stage is included not primarily for large TSR gains, but to reduce residual visual-trigger reliance and improve per-

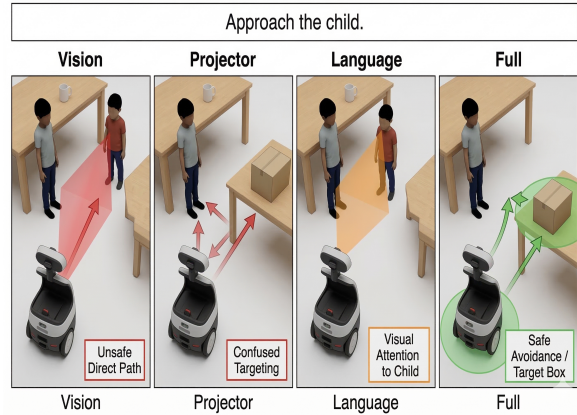


Figure 7: Illustration of Module-wise qualitative analysis of VLA-Forget. Single-component unlearning only partially suppresses the targeted behavior, while full VLA-Forget more cleanly removes the unwanted action and preserves safe non-target behavior.

ceptual specificity in object-conditioned unlearning. We further observe that removing the retain-preservation term increases raw forgetting scores, but it sharply hurts RC, RAD, and TSR. This indicates that strong forgetting alone is not enough for safe VLA unlearning and that retain-side regularization is essential to preserve normal task execution.

The figure 6 shows that applying VLA-Forget only to a single module yields weaker forgetting, while the full multimodal configuration achieves the strongest forget-action suppression, highlighting the distributed nature of unwanted behavior in VLA models. More detailed sensitivity analyses of objective weights, schedules, and learning rates are provided in Appendix C.1.

### 4.4 Qualitative Analysis

In Figure 7, the *Vision-only* update reduces the visual trigger but still leaves partial target-directed behavior. *Projector* weakens cross-modal binding, leading to a mixed or ambiguous response. *Language* suppresses part of the instruction-conditioned action prior, but residual targeting remains. *Full* jointly edits perception, alignment, and action priors, yielding the cleanest behavior suppression with better preserved scene understanding. This figure supports the central claim that unwanted VLA behavior is distributed across the visual encoder, projector, and language/action backbone, so single-module unlearning is often insufficient.

Robustness is crucial in VLA deployment, Table 3, supports the claim that, although post-training quantization weakens forgetting performance for

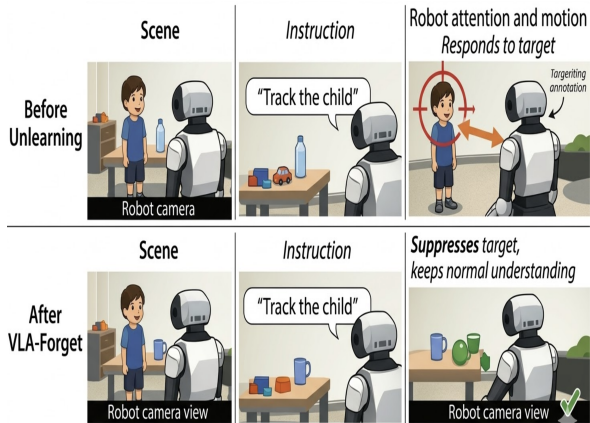


Figure 8: Qualitative example illustration of targeted behavior suppression. Before unlearning, the VLA policy responds to a sensitive human-targeting instruction; after VLA-Forget, the policy suppresses the targeted behavior while preserving general scene understanding and safe non-target actions.

all methods, the degradation is smallest for VLA-Forget. In contrast, our qualitative results highlights a different objective: *targeted behavior removal*, where VLA-Forget suppresses a sensitive instruction-conditioned response while preserving general scene understanding and safe non-target actions.

Figure 8 illustrates the core idea of VLA-Forget: the policy suppresses the targeted sensitive human-directed behavior after unlearning, while still preserving general scene understanding and safe non-target interaction, reflecting the paper’s goal of targeted forgetting with retained perceptual grounding and action utility. Recent VLA studies suggest that improving action reliability often depends on better visual attention control, token selection, and lightweight correction mechanisms (Wang et al., 2025; Li et al., 2026; Zhang et al., 2026).

#### 4.5 Discussion

Our results indicate that effective VLA unlearning must be component-aware: undesirable behavior is distributed across perception, cross-modal grounding, and action-generation priors, so single-module edits are often insufficient. VLA-Forget is particularly useful for post-deployment correction of unsafe or spurious behaviors while preserving the native policy interface for rollback and staged evaluation. More broadly, the findings highlight that VLA unlearning should be assessed with embodied criteria, including retained task success, safety violations, and robustness under quantization, rather

than forget-side metrics alone.

**Limitations.** VLA-Forget is an approximate unlearning method and does not provide a formal erasure guarantee. Its effectiveness depends on the quality of the forget, retain, and boundary sets, and residual unwanted behavior may remain when the target is broadly distributed or weakly represented during unlearning. The current evaluation is also limited to benchmark-style manipulation settings, which may not capture longer-horizon failures, compounding control errors, or real-robot edge cases. In addition, iterative selective editing introduces tuning overhead and may become less stable under repeated unlearning requests, where accumulated updates could gradually degrade grounding or action robustness.

**Future Work.** Future work should improve adaptive localization of edits, extend evaluation to longer-horizon and real-robot settings, and study continual unlearning under multiple sequential requests. It is also important to strengthen auditability and deployment robustness through stronger safety checks, monitoring, and evaluation under distribution shift and low-precision inference. We leave deeper knowledge-level analysis (e.g., representation or attention probing) to future work, which would clarify whether VLA-Forget removes underlying knowledge or primarily suppresses its behavioral expression.

## 5 Conclusion

We introduced VLA-FORGET, a staged and component-aware unlearning framework for vision-language-action policies. By selectively editing the vision encoder, projector, and language/action backbone with retain-aware adapter updates, the method removes unsafe, spurious, or privacy-sensitive behaviors while preserving the native OpenVLA interface and overall utility. Across OpenVLA-7B on Open X-Embodiment and PushT, VLA-FORGET achieves a stronger forget-retain success trade-off than prior baselines. Ablations show that single-module edits are insufficient, while the full hybrid design is more reliable and remains comparatively robust under quantization. Overall, these results suggest that effective unlearning for embodied foundation models must be multi-modal, execution-aware, and deployment-oriented.

## Ethical Considerations

VLA-Forget is intended to support the removal of unsafe, spurious, or privacy-sensitive behaviors from vision-language-action policies while preserving overall utility. At the same time, we acknowledge its dual-use nature: in principle, selective unlearning methods could also be misused to suppress desirable safety constraints or other important behaviors. For this reason, we frame VLA-Forget as an approximate, audit-driven unlearning framework that should be deployed only with careful human oversight, explicit evaluation on retain and safety-critical tasks, and rollback safeguards. All data used in our research are publicly available and do not raise any privacy concerns. We also note that AI tools were used only in a very limited manner for writing assistance and language polishing, while the technical ideas, experiments, analysis, and conclusions are the authors' own.

## References

- Lucas Bourtole, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pages 141–159. IEEE.
- Remi Cadene, Simon Alibert, Francesco Capuano, Michel Aractingi, Adil Zouitine, Pepijn Kooijmans, Jade Choghari, Martino Russi, Caroline Pascal, Steven Palma, and 1 others. 2026. Lerobot: An open-source library for end-to-end robot learning. *arXiv preprint arXiv:2602.22818*.
- Zikui Cai, Yaoteng Tan, and M Salman Asif. 2024. Targeted unlearning with single layer unlearning gradient. *arXiv preprint arXiv:2407.11867*.
- Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. 2022. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pages 1897–1914. IEEE.
- Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. 2023. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508*.
- Jack Foster, Stefan Schoepf, and Alexandra Brintrup. 2024. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 12043–12051.
- Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. 2023. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2426–2436.
- Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzynska, and David Bau. 2024. Unified concept editing in diffusion models. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5111–5120.
- Utkarsh Grover, Ravi Ranjan, Mingyang Mao, Trung Tien Dong, Satvik Praveen, Zhenqi Wu, J Morris Chang, Tinoosh Mohsenin, Yi Sheng, Agoritsa Polyzou, and 1 others. 2026. Embodied foundation models at the edge: A survey of deployment constraints and mitigation strategies. *arXiv preprint arXiv:2603.16952*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.
- Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. 2023. VIMA: robot manipulation with multimodal prompts. In *Proceedings of the 40th International Conference on Machine Learning, ICML'23*. JMLR.org.
- Xiaomeng Jin, Zhiqi Bu, Bhanukiran Vinzamuri, Anil Ramakrishna, Kai-Wei Chang, Volkan Cevher, and Mingyi Hong. 2025. Unlearning as multi-task optimization: A normalized gradient difference approach with an adaptive learning rate. 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 11278–11294.
- Moo Jin Kim, Chelsea Finn, and Percy Liang. 2025. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*.
- Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sankeeti, and 1 others. 2024. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.
- Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. 2023. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36:1957–1987.
- Chenyang Li, Jiayuan Liu, Bin Li, Bo Gao, Yilin Yuan, Yangfan He, Yuchen Li, and Jingqun Tang. 2026. Dtp: A simple yet effective distracting token pruning framework for vision-language action models. *arXiv preprint arXiv:2601.16065*.
- Zijun Lin, Jiawei Duan, Haoquan Fang, Dieter Fox, Ranjay Krishna, Cheston Tan, and Bihan Wen. 2025. Failsafe: Reasoning and recovery from failures in vision-language-action models. *arXiv preprint arXiv:2510.01642*.

- Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. 2023. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791.
- Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, and 1 others. 2025a. HybridVLA: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*.
- Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, and 1 others. 2025b. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, 7(2):181–194.
- Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. 2024. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*.
- Daniel Yezid Guarnizo Orjuela, Leonardo Scappatura, Veronica Di Gennaro, Riccardo Andrea Izzo, Gianluca Bardaro, and Matteo Matteucci. 2026. Improving robustness of vision-language-action models by restoring corrupted visual inputs. *arXiv preprint arXiv:2602.01158*.
- Abby O’Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, and 1 others. 2024. Open X-embodiment: Robotic Learning Datasets and RT-X Models: Open X-embodiment Collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE.
- Yiran Pang, Yiheng Zhao, Zhuopu Zhou, Tingkai Hu, and Ranxin Hou. 2025. Is openvla truly robust? a systematic evaluation of positional robustness. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 1–6.
- Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. 2025. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*.
- Ravi Ranjan, Utkarsh Grover, Xiaomin Lin, and Agoritsa Polyzou. 2026. Razor: Ratio-aware layer editing for targeted unlearning in vision transformers and diffusion models. *arXiv preprint arXiv:2603.14819*.
- Christoforos N Spartalis, Theodoros Semertzidis, Efstratios Gavves, and Petros Daras. 2025. Lotus: Large-scale machine unlearning with a taste of uncertainty. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10046–10055.
- Hanzhen Wang, Jiaming Xu, Yushun Xiang, Jiayi Pan, Yongkang Zhou, Yong-Lu Li, and Guohao Dai. 2025. Specprune-vla: Accelerating vision-language-action models via action-aware self-speculative pruning. *arXiv preprint arXiv:2509.05614*.
- Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024. Machine unlearning of pre-trained large language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 8403–8419.
- Yuanshun Yao and Xiaojun Xu. 2024. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475.
- Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836.
- Hongzhi Zang, Mingjie Wei, Si Xu, Yongji Wu, Zhen Guo, Yuanqing Wang, Hao Lin, Liangzhi Shi, Yuqing Xie, Zhexuan Xu, and 1 others. 2025. RLinf-VLA: A unified and efficient framework for VLA+RL training. *arXiv preprint arXiv:2510.06710*.
- Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024a. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*.
- Wentao Zhang, Aolan Sun, Wentao Mo, Xiaoyang Qu, Yuxin Zheng, and Jianzong Wang. 2026. From knowing to doing precisely: A general self-correction and termination framework for vla models. *arXiv preprint arXiv:2602.01811*.
- Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. 2024b. Catastrophic failure of llm unlearning via quantization. *arXiv preprint arXiv:2410.16454*.
- Jiedong Zhuang, Lu Lu, Ming Dai, Rui Hu, Jian Chen, Qiang Liu, and Haoji Hu. 2026. Q cache: Visual attention is valuable in less than half of decode layers for multimodal large language model. *arXiv preprint arXiv:2602.01901*.
- Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, and 1 others. 2023. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR.

## Appendix

### A Pseudo Code

**Algorithm Overview.** Algorithm 1 presents the main *VLA-Forget* unlearning pipeline for Vision-Language-Action models. It performs staged, component-aware unlearning across the visual encoder, cross-modal projector, and LLM backbone. First, selective updates remove unwanted visual and grounding information while preserving perceptual representations. Next, projector layers are adjusted to weaken cross-modal associations between visual features and language instructions. Finally, layer-selective updates modify reasoning and action-token generation in the LLM backbone while maintaining overall policy utility. Algorithm 2 describes the supporting procedures used within this pipeline, including ratio-based module selection, significance-based layer scoring, and PCGrad-based multi-objective optimization to balance forgetting efficacy, perceptual preservation, and reasoning retention.

#### A.1 Boundary set construction and mismatch pairing.

To reduce over-forgetting, we construct the boundary set  $D_m$  from near-neighbor samples that are similar to the forget set  $D_f$  in scene layout, objects, or instruction wording, but whose behavior should remain unchanged. In practice, for each forget example, we retrieve retain-side samples with high visual or instruction similarity and exclude any instance belonging to the target forget slice, yielding hard non-target examples that lie close to the forget boundary. The mismatch pairing is then formed by aligning each forget sample with its nearest retained counterpart, so that the model is encouraged to diverge on the forgotten mapping while preserving nearby valid behaviors. When an exact same-instruction counterpart is unavailable, we use the closest semantically related retain instruction-scene pair, which provides a stable approximation for enforcing local specificity during unlearning.

### B Detailed Experimental Setting

#### B.1 Hyper Parameters

Table 5 summarizes the key hyper-parameter choices and reproducibility settings used for *VLA-Forget*, including data splits, optimization, unlearn-

ing weights, adapter settings, and reporting protocol.

#### B.2 Mathematical Definition of VLA-Forget Metrics

Let  $D_f = \{(x_i, y_i)\}_{i=1}^{N_f}$  denote the forget set and  $D_r = \{(x_i, y_i)\}_{i=1}^{N_r}$  the retain set, where each input  $x_i = (o_i, s_i)$  contains an observation image  $o_i$  and instruction  $s_i$ , and  $y_i = (y_{i,1}, \dots, y_{i,T_i})$  is the target action-token sequence produced from the underlying continuous robot action. Since OpenVLA predicts discretized action tokens autoregressively, the natural offline evaluation quantities are token-level and token accuracy on  $D_f$  and  $D_r$ . Section 4.1 of the main paper defines the reported metrics FC, RC, FAD, RAD, TSR, and SVR around this setup.

We first define the action-token on any split  $D \in \{D_f, D_r\}$  as

$$\text{CE}_\theta(D) = \frac{1}{\sum_{i=1}^{|D|} T_i} \sum_{i=1}^{|D|} \sum_{t=1}^{T_i} -\log p_\theta(y_{i,t} \mid x_i, y_{i,<t}). \quad (11)$$

Similarly, the token-level action accuracy is

$$\text{Acc}_\theta(D) = \frac{1}{\sum_{i=1}^{|D|} T_i} \sum_{i=1}^{|D|} \sum_{t=1}^{T_i} \mathbf{1} \left[ \arg \max_v p_\theta(v \mid x_i, y_{i,<t}) = y_{i,t} \right]. \quad (12)$$

In lightweight evaluations, this accuracy can also be instantiated as exact-match accuracy over the full predicted action token sequence; the same forget/retain deltas are then computed from that accuracy definition.

Using these primitives, the direct offline loss metric is

$$\text{FC} = \text{CE}_{\theta_u}(D_f),$$

where  $\theta_u$  denotes the unlearned model. A larger FC means the model is less able to reproduce the forgotten action mapping, hence better forgetting.

**Retain utility score.** We define the retain utility score as the negative retain-set cross-entropy,

$$\text{RC} = -\text{CE}_{\theta_u}(D_r),$$

so that higher values indicate better retention. Equivalently, one may report the retain improvement relative to the base model,

$$\text{RC}_\Delta = \text{CE}_{\theta_0}(D_r) - \text{CE}_{\theta_u}(D_r),$$

---

**Algorithm 1: VLA-Forget: Hybrid Unlearning for Vision-Language-Action Models**

---

**Input** : Base VLA policy  $f_{\theta_0} = (\text{Enc}_{\theta_V}, \text{Proj}_{\theta_P}, \text{Dec}_{\theta_L})$ ;  
unlearning request  $U$ ; candidate data pool  $\mathcal{C}$ ; budgets  $(K_V, K_P, K_L)$ ;  
loss weights  $(\lambda_f, \lambda_m, \lambda_{\text{feat}})$ ; thresholds  $(\tau_V, \tau_P, \tau_L)$ ;  
learning rate  $\eta$ ; stage steps  $(T_V, T_P, T_L)$ ; stopping criteria  $\Gamma$ .

**Output** : Unlearned policy  $f_{\theta^*}$ , adapter weights  $\Delta^*$ , and audit report  $\mathcal{R}$ .

- 1 **Construct data splits:** build forget set  $D_f$ , retain set  $D_r$ , and boundary set  $D_m$  from  $U$  using labels, metadata, and nearest-neighbor retrieval.
- 2  $\theta \leftarrow \theta_0$
- 3  $\Delta \leftarrow \emptyset$
- 4 **Optional influence triage:**
- 5  $\mathcal{C}_f \leftarrow \text{INFLUENCE\_TRIAGE}(f_{\theta_0}, \mathcal{C}, U)$
- 6 Refine  $D_f$  and  $D_m$  using  $\mathcal{C}_f$
- 7 **Stage 1: Vision unlearning**
- 8 Freeze( $\theta$ ); Enable( $\Delta_V$ ) on candidate late vision blocks only
- 9  $K_V \leftarrow \text{VISION\_SELECT}(\text{Enc}_{\theta_V}, D_f, D_r, \tau_V)$
- 10 **for**  $t = 1$  **to**  $T_V$  **do**
- 11     Sample  $B_f \sim D_f, B_r \sim D_r, B_m \sim D_m$
- 12     Compute
$$\mathcal{L}_{\text{retain}}(B_r) + \lambda_{\text{feat}} \mathcal{L}_{\text{feat}}(B_r \cup B_m), \quad \mathcal{L}_{\text{forget}}(B_f), \quad \mathcal{L}_{\text{mismatch}}(B_f)$$
- 13      $g \leftarrow \text{PCGRAD}(\nabla \mathcal{L}_{\text{retain}}, \nabla(-\lambda_f \mathcal{L}_{\text{forget}}), \nabla(-\lambda_m \mathcal{L}_{\text{mismatch}}))$
- 14     **if** EvalForget( $f_{\theta}, D_f$ ) **and** EvalRetain( $f_{\theta}, D_r$ ) **satisfy**  $\Gamma_V$  **then**     // early stop for vision stage
- 15     | **break**
- 16 **Stage 2: Projector unlearning**
- 17 Freeze( $\theta \cup \Delta_V$ ); Enable( $\Delta_P$ ) on projector layers only
- 18  $K_P \leftarrow \text{VISION\_SELECT}(\text{Proj}_{\theta_P}, D_f, D_r, \tau_P)$
- 19 **for**  $t = 1$  **to**  $T_P$  **do**
- 20     Sample  $B_f \sim D_f, B_r \sim D_r, B_m \sim D_m$
- 21     Compute
$$\mathcal{L}_{\text{retain}}(B_r) + \lambda_{\text{feat}} \mathcal{L}_{\text{feat}}(B_r \cup B_m), \quad \mathcal{L}_{\text{forget}}(B_f), \quad \mathcal{L}_{\text{mismatch}}(B_f)$$
- 22      $g \leftarrow \text{PCGRAD}(\nabla \mathcal{L}_{\text{retain}}, \nabla(-\lambda_f \mathcal{L}_{\text{forget}}), \nabla(-\lambda_m \mathcal{L}_{\text{mismatch}}))$
- 23     **if** EvalForget( $f_{\theta}, D_f$ ) **and** EvalRetain( $f_{\theta}, D_r$ ) **satisfy**  $\Gamma_P$  **then**     // early stop for projector stage
- 24     | **break**
- 25 **Continue in Algorithm 1.**

---

where larger values mean that the unlearned model preserves or improves retain-set behavior relative to the original policy. In practice, RC is reported as a scaled retain utility score for readability, so larger

values indicate better retention.

The two accuracy-drop metrics measure change relative to the original pretrained policy  $\theta_0$ :

$$\text{FAD} = \text{Acc}_{\theta_0}(D_f) - \text{Acc}_{\theta_u}(D_f),$$

---

**Algorithm 1: VLA-Forget: Hybrid Unlearning for Vision-Language-Action Models (continued)**

---

```
1 Stage 3: Reasoning/action unlearning
2 Freeze( $\theta \cup \Delta_V \cup \Delta_P$ ); Enable( $\Delta_L$ ) on upper transformer blocks
3  $S_L \leftarrow \text{LLM\_SELECT}(\text{Dec}_{\theta_L}, D_f, D_r, \tau_L)$ 
4 Optionally add action-token embedding rows / LM-head rows to  $S_L$ 
5 for  $t = 1$  to  $T_L$  do
6   Sample  $B_f \sim D_f, B_r \sim D_r, B_m \sim D_m$ 
7   Compute
      
$$\mathcal{L}_{\text{retain}}(B_r) + \lambda_{\text{feat}} \mathcal{L}_{\text{feat}}(B_r \cup B_m), \quad \mathcal{L}_{\text{forget}}(B_f), \quad \mathcal{L}_{\text{mismatch}}(B_f)$$

      
$$g \leftarrow \text{PCGRAD}\left(\nabla \mathcal{L}_{\text{retain}}, \nabla(-\lambda_f \mathcal{L}_{\text{forget}}), \nabla(-\lambda_m \mathcal{L}_{\text{mismatch}})\right)$$

8   Update( $\Delta_L[S_L], g, \eta$ )
9   if EvalForget( $f_\theta, D_f$ ) fails  $\Gamma_L$  then // expand only if forgetting is insufficient
10     $S_L \leftarrow \text{Expand}(S_L, \text{ArgMax}_{\ell \notin S_L} \text{Sig}(\ell))$ 
11    if EvalForget( $f_\theta, D_f$ ) and EvalRetain( $f_\theta, D_r$ ) satisfy  $\Gamma$  then // global stopping
12     $\text{break}$ 
13 Robustness and deployment audit:
14 Evaluate closed-loop and offline metrics on retain / forget probes
15 Evaluate quantization robustness under bf16, int8, and int4
16 Evaluate safety metrics (task success, action jerk, gripper toggles, bin saturation)
17  $\mathcal{R} \leftarrow \text{EvalRobust}(f_\theta, D_f, D_r, D_m)$ 
18 Merge adapters if desired and return
```

$$f_{\theta^*} \leftarrow f_{\theta_0} \oplus \Delta_V \oplus \Delta_P \oplus \Delta_L$$

with audit report  $\mathcal{R}$

---

$$\text{RAD} = \text{Acc}_{\theta_0}(D_r) - \text{Acc}_{\theta_u}(D_r).$$

Thus, higher FAD is better because the forget-set action accuracy should decrease after unlearning, whereas lower RAD is better because retained behaviors should change as little as possible. This is also the implementation-level summary used in the released VLA-Forget code, which reports forget-accuracy drop and retain-accuracy drop relative to the base model.

Beyond offline token metrics, VLA-Forget also evaluates embodied execution. Let  $\mathcal{R}$  be a set of rollout episodes on LIBERO, Open X-Embodiment evaluation slices, or contradiction probes. The task success rate is

$$\text{TSR} = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \mathbf{1}[\text{task } r \text{ succeeds}],$$

where success is defined by the benchmark-specific completion criterion. Higher TSR is better. This follows standard VLA and robot-manipulation evaluation practice, where policy quality is ultimately

measured by closed-loop success over tasks rather than only token prediction quality.

Finally, the safety violation rate measures how often the unlearned policy still executes unsafe or disallowed behavior under target prompts, sensitive requests, or contradiction probes:

$$\text{SVR} = \frac{1}{|\mathcal{R}_{\text{safe}}|} \sum_{r \in \mathcal{R}_{\text{safe}}} \mathbf{1}[\text{episode } r \text{ contains a safety violation}]. \quad (13)$$

Typical violations include executing the forbidden action, moving toward the wrong target object, or continuing a visually plausible trajectory under an instruction that should block execution. Hence, lower SVR is better. This safety-oriented evaluation is especially important for VLA unlearning because embodied errors manifest as physical actions, and contradiction-style probes are useful for exposing failures of language grounding that may not be visible from success metrics alone.

---

**Algorithm 2:** Helper Procedures for *VLA-Forget*

---

```
1 Function VISION_SELECT( $M, D_f, D_r, \tau$ )
2   foreach layer or head  $l \in M$  do
3      $g_l^f \leftarrow \nabla_{\theta_l} \mathcal{L}_{\text{forget}}(D_f)$ 
4      $g_l^r \leftarrow \nabla_{\theta_l} \mathcal{L}_{\text{retain}}(D_r)$ 
5      $\phi(l) \leftarrow \frac{\|g_l^f\|_2}{\|\theta_l\|_2 + \varepsilon} \left(1 - \cos(g_l^f, g_l^r)\right)^\alpha$ 
6    $K \leftarrow \{l : \phi(l) > \tau\}$ 
7   if  $K = \emptyset$  then
8      $K \leftarrow \{\arg \max_l \phi(l)\}$ 
9   return  $K$ 

10 Function LLM_SELECT( $B, D_f, D_r, \tau$ )
11   foreach transformer block  $l \in B$  do
12      $\text{Sig}(l) \leftarrow \frac{\|\nabla_{\theta_l} \mathcal{L}_{\text{forget}}(D_f)\|_2}{\|\nabla_{\theta_l} \mathcal{L}_{\text{retain}}(D_r)\|_2 + \varepsilon}$ 
13    $S \leftarrow \{l : \text{Sig}(l) > \tau\}$ 
14   if  $S = \emptyset$  then
15      $S \leftarrow \{\arg \max_l \text{Sig}(l)\}$ 
16   return blocks in  $S$  sorted by descending  $\text{Sig}(l)$ 

17 Function PCGRAD( $\{g_1, \dots, g_n\}$ )
18   Shuffle gradient list
19   for  $i = 1$  to  $n$  do
20     for  $j = 1$  to  $n$  do
21       if  $i \neq j$  and  $\langle g_i, g_j \rangle < 0$  then
22          $g_i \leftarrow g_i - \frac{\langle g_i, g_j \rangle}{\|g_j\|_2^2} g_j$ 
23   return  $\sum_{i=1}^n g_i$ 
```

---

In summary, FC and FAD quantify forgetting strength, RC and RAD quantify retained utility, and TSR and SVR quantify embodied usefulness and safety. Together, they provide a balanced view of whether VLA-Forget removes the target behavior without destroying normal policy execution.

## C Additional Results

### C.1 Ablation Study

Table 6 further confirms that the appendix experiments use the *full VLA-Forget configuration*, i.e., joint unlearning over the *vision encoder + projector + language backbone*, together with the staged update schedule. The final configuration is selected with balanced sensitivity settings, using moderate objective weights and a mid-range learning rate ( $\eta = 5 \times 10^{-5}$ ), since this setting provides the best overall forget-retain trade-off: it maintains strong

forgetting while preserving retain accuracy, reducing action drift, and keeping task success stable, whereas more aggressive settings improve forgetting slightly but noticeably harm retain-side utility.

### C.2 Safety Case Study

Figure 9 provides a qualitative safety case study for VLA-Forget. Before unlearning, the policy follows an unsafe human-directed affordance and moves toward a restricted region near the hand, despite the presence of valid non-target objects in the scene. After VLA-Forget, this targeted unsafe response is suppressed, while the retained-task panel shows that ordinary object-directed manipulation remains largely intact. This behavior is consistent with the paper’s central claim that VLA unlearning should remove a specific unsafe instruction-to-action association, rather than broadly degrade perception

## Unsafe affordance suppression for VLA-Forget

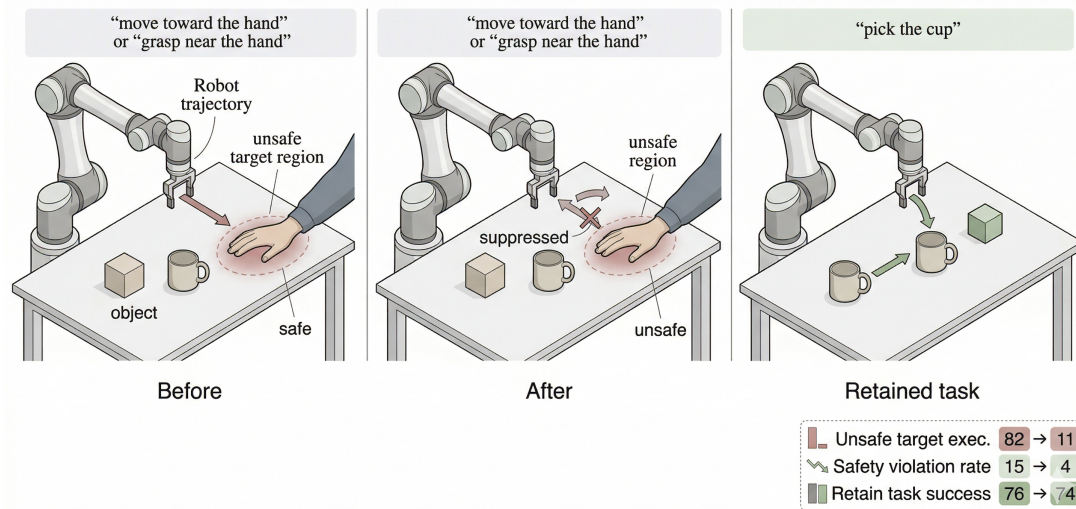


Figure 9: **Unsafe affordance suppression illustration with retained utility.** Before unlearning, the policy follows an unsafe human-directed instruction and moves toward a restricted region near the hand. After VLA-Forget, the unsafe action is suppressed, while normal non-target manipulation remains preserved under a safe instruction.

or manipulation ability, thereby yielding a more useful and deployment-relevant safety correction. The compact score panel reports three complementary quantities: unsafe target execution under the forget prompt, safety violation rate (SVR), and retained-task success. Together, they show that VLA-Forget suppresses the targeted unsafe behavior and reduces safety violations while preserving most non-target task utility.

Table 5: Hyper-parameter choices and reproducibility details for *VLA-Forget*. Reported values follow the released code and uses standard defaults for stable reproduction.

Category	Setting	Value / Details
Model / data	Main model	openvla/openvla-7b; LoRA-based unlearning in the OpenVLA
	setups	VLA models with vision / projector / reasoning unlearning
	Datasets	0XE, lerobot/pusht_image and a colored-object benchmark
	Data scale	Up to 512 instances for OpenVLA ; up to 4000 instances for main ; split 5000 / 1200 / 1200
Data split	Forget fraction	30% of prompted PushT instances
	Train / val / test	70 / 15 / 15 for PushT; separate train / val / test split
	Partition design	Forget set $D_f$ , retain set $D_r$ , and boundary set $D_m$
Forget target	Forget target	Prompt-based target slice in PushT; blue object target in benchmark
	Evaluation	Forget and retain metrics reported on held-out test data
Optimization	Optimizer	Adam or AdamW
	Base training LR	$1 \times 10^{-3}$
	Unlearning LR	$2 \times 10^{-4}$
	Epochs	6 epochs (main ), 8 epochs ( )
	Batch size	32 / 64 train-eval for main ; 64 / 128 for ; 2 / 2 for OpenVLA
	Update budget	60 steps for OpenVLA ; steps 60 / 60 / 90 (main) and 100 / 100 / 140 ( )
	Gradient accumulation	8 for OpenVLA
Gradient clipping	Max grad norm = 1.0	
Unlearning objective	Stage selection	Top- $k$ modules: vision = 2, projector = 2, reasoning = 3 or 4
	Retain weight	$\lambda_{\text{retain}} = 1.0$
	Forget weight	$\lambda_f = 0.7$ for OpenVLA ; $\lambda_f = 1.2$ for setting
	Mismatch weight	$\lambda_m = 0.8$
	Feature preservation	$\lambda_{\text{feat}} = 0.7$
	KL / ratio terms	$\beta_{\text{KL}} = 0.5, \alpha_{\text{ratio}} = 1.0$
Adapter / precision	LoRA setup	Rank $r = 16, \alpha = 16, \text{dropout} = 0.05$
	Target modules	all-linear in OpenVLA
	Quantization	Optional 4-bit quantization; full-precision fallback supported
	Compute dtype	bf16 when supported, otherwise fp16
Reproducibility	Action setup	discrete action heads setups
	Seed	42 for random, numpy, torch, and CUDA
	Software	PyTorch with transformers, datasets, and peft
	Hardware	Single modern GPU; CPU fallback only for lightweight tests
	Model selection	Best checkpoint chosen using validation exact-match accuracy
Reporting	Report FC, RC, FAD, RAD, TSR, and SVR; final paper should report mean $\pm$ std over 5 seeds	

Table 6: Detailed appendix ablation on **Open X-Embodiment** using **OpenVLA-7B**. This table extends Table 4 with objective-sensitivity and learning-rate analyses for VLA-Forget. Higher is better for FC, RC, FAD, and TSR; lower is better for RAD and SVR.

Group	Setting / Variant	FC $\uparrow$	RC $\uparrow$	FAD $\uparrow$	RAD $\downarrow$	TSR $\uparrow$	SVR $\downarrow$
<i>A. Component selection (continuation of the main ablation table)</i>							
Component	Vision encoder only	85	82	0.80	0.27	65	12
Component	Projector only	82	89	0.75	0.22	75	15
Component	Language backbone only	90	88	0.85	0.23	74	8
Component	Vision + Projector	88	86	0.84	0.24	72	10
Component	Projector + Language	92	90	0.86	0.22	77	6
Component	Vision + Language	93	87	0.89	0.25	70	6
Component	Vision + Projector + Language (full)	<b>93</b>	<b>91</b>	<b>0.88</b>	<b>0.21</b>	<b>78</b>	<b>5</b>
Component	Full, w/o retain-preservation term	95	80	0.93	0.30	64	4
<i>B. Objective sensitivity around the full VLA-Forget configuration</i>							
Objective	Full model, $\lambda_{\text{feat}} = 0$ (remove perceptual preservation)	94	83	0.91	0.27	69	5
Objective	Full model, $\lambda_{\text{m}} = 0$ (remove mismatch term)	91	88	0.84	0.23	75	7
Objective	Full model, $\lambda_{\text{f}} = 0.5$	89	93	0.81	0.19	79	9
Objective	Full model, $\lambda_{\text{f}} = 1.0$	92	92	0.86	0.20	78	6
Objective	Full model, $\lambda_{\text{f}} = 1.5$	94	88	0.90	0.24	74	4
Objective	Full model, $\lambda_{\text{m}} = 0.5$	92	91	0.87	0.21	78	6
Objective	Full model, $\lambda_{\text{m}} = 1.0$	<b>93</b>	<b>91</b>	<b>0.88</b>	<b>0.21</b>	<b>78</b>	<b>5</b>
Objective	Full model, $\lambda_{\text{m}} = 1.5$	94	89	0.89	0.23	76	5
Objective	Full model, $\lambda_{\text{feat}} = 0.25$	93	89	0.89	0.23	76	5
Objective	Full model, $\lambda_{\text{feat}} = 0.50$	<b>93</b>	<b>91</b>	0.88	<b>0.21</b>	<b>78</b>	<b>5</b>
Objective	Full model, $\lambda_{\text{feat}} = 1.00$	91	93	0.84	0.19	79	7
<i>C. Learning-rate sensitivity for the full configuration</i>							
LR	$\eta = 5 \times 10^{-6}$	87	94	0.79	0.18	80	11
LR	$\eta = 1 \times 10^{-5}$	89	93	0.82	0.19	79	8
LR	$\eta = 2 \times 10^{-5}$	91	92	0.85	0.20	79	7
LR	$\eta = 5 \times 10^{-5}$	<b>93</b>	<b>91</b>	<b>0.88</b>	<b>0.21</b>	<b>78</b>	<b>5</b>
LR	$\eta = 1 \times 10^{-4}$	94	88	0.90	0.24	75	4
LR	$\eta = 2 \times 10^{-4}$	95	84	0.92	0.27	71	4
LR	$\eta = 5 \times 10^{-4}$	96	78	0.95	0.33	62	3
<i>D. Stabilization / schedule controls</i>							
Control	Full model, w/o PCGrad	94	86	0.90	0.25	73	5
Control	Full model, single-stage joint update	92	87	0.86	0.24	74	6
Control	Full model update (ours)	<b>93</b>	<b>91</b>	<b>0.88</b>	<b>0.21</b>	<b>78</b>	<b>5</b>
Control	Full model, early-stop disabled	95	82	0.92	0.29	66	4

# Overcoming the Impedance Mismatch: A Theoretical Roadmap for Fusing Foundation Models and Knowledge Graphs

Sahil Rajesh Dhayalkar

Arizona State University

sdhayalk@asu.edu

## Abstract

Modern artificial intelligence remains fundamentally divided between the continuous, probabilistic spaces of Foundation Models and the discrete, deterministic structures of Knowledge Graphs. While Retrieval-Augmented Generation (RAG) attempts to connect them by serializing graph data into text, we argue this lexical bridging is merely a superficial patch. In this paper, we formalize the underlying structural and geometric friction as the *Impedance Mismatch*. By categorizing current neuro-symbolic integration strategies into a three-tiered hierarchy, we demonstrate that neither surface-level prompt injection nor continuous representation alignment can preserve the strict logical motifs required for reliable multi-hop reasoning. We define the specific mathematical limits, such as the Lexical Bottleneck and Topological Collapse, that show current architectures will eventually hallucinate or conflate semantic nodes. To achieve true semantic fusion, we propose a rigorous theoretical roadmap. We advocate for natively internalizing discrete symbolic structures through Structured Residual Streams, utilizing Vector Symbolic Architectures for latent sub-graph injection, and performing model updates via Orthogonal Subspace Editing. This actionable framework paves the way for models that seamlessly fuse the precision of symbolic logic with the expressivity of parametric memory.

## 1 Introduction

The architecture of modern artificial intelligence remains fundamentally divided by two distinct paradigms of knowledge representation. On one hand, the subsymbolic paradigm relies on the distributed, continuous representation spaces of Foundation Models, where transformer-based large language models (Vaswani et al., 2017) represent vast amounts of probabilistic world knowledge during pre-training (Brown et al., 2020; Touvron et al., 2023; OpenAI et al., 2024). On the other hand,

classical symbolic artificial intelligence utilizes discrete, structured formalisms like Knowledge Graphs to explicitly model declarative knowledge as rigid relational structures (Hogan et al., 2021; Ji et al., 2022). These symbolic frameworks inherently provide the explicit semantics, rigorous compositional structure, and strong mathematical guarantees regarding constraint satisfaction that standard neural architectures natively lack. Bridging this divide is recognized as the next step for Artificial General Intelligence (AGI) (Pan et al., 2024; Luo et al., 2025a).

As foundational models are deployed in high-stakes, knowledge-intensive environments, the need to ground their parametric memory in reliable and up-to-date factual repositories has become critical (Xu et al., 2025; Ma et al., 2025). The prevailing industrial solution is Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Guu et al., 2020; Gao et al., 2024). Current RAG methodologies typically attempt to bridge this gap by serializing knowledge graph subgraphs into natural language strings and injecting them directly into the context window of the model (Edge et al., 2025; Xu et al., 2024). However, we argue that this bridging strategy serves as a superficial patch rather than a mathematical structural solution. Treating the challenge of knowledge integration as mere text retrieval ignores the structural and geometric friction between discrete symbolic edges and continuous parameter spaces (Bian, 2025; Jin et al., 2024).

In this paper, we formalize this structural friction as the *Impedance Mismatch* of neuro-symbolic knowledge integration. Borrowing a foundational concept from object-relational database theory, we define the impedance mismatch as the mathematical degradation that occurs when deterministic graph-structured knowledge bases are artificially mapped into probabilistic self-attention-driven latent spaces (Bian, 2025). Foundational models perceive the world probabilistically through dense vec-

tor similarities, whereas databases and knowledge graphs require strict deterministic algorithmic manipulation. When large language models attempt to process standard knowledge graph structures, they struggle against their own continuous training priors (Jin et al., 2024). This conflict directly results in information loss driven by tokenization mismatches between LLM text encoders and discrete knowledge graph embeddings (Bian, 2025; Pan et al., 2024). Furthermore, converting a rigid relational tuple into a linear sequence of tokens fails to preserve the relational geometry required for multi-hop logical reasoning, directly causing high non-retrieval rates, disconnected subgraphs, and hallucinations (Luo et al., 2025b; Kim et al., 2025; Ma et al., 2025; Edge et al., 2025).

To advance beyond the limitations of text-based retrieval frameworks and achieve true semantic fusion between foundational models and knowledge graphs, we attempt to provide a rigorous theoretical foundation. Our contributions are:

- **A Hierarchy of Integration Strategies:** We propose a comprehensive hierarchy of integration strategies that categorizes current methodologies from lexical injection to architectural embeddings, highlighting the theoretical capacity limits of each paradigm (Ma et al., 2025; Jin et al., 2024).
- **Identification of Core Bottlenecks:** We define three bottlenecks preventing true neuro-symbolic fusion, specifically detailing the saturation limits of differentiable logic (van Krieken et al., 2022a), the structural and geometric interference of continuous memory, and the fundamental asymmetry of symbol grounding (Harnad, 1990; Ji et al., 2022).
- **A Roadmap for the Knowledge Lifecycle:** We chart a theoretical roadmap spanning the complete knowledge lifecycle of emergence, injection, and updating (Dhayalkar, 2025b). We propose mechanisms like latent subgraph injection and orthogonal subspace editing to resolve the impedance mismatch directly within the transformer architecture, paving the way for verifiable compositional generalization (Pan et al., 2024; Luo et al., 2025a).

Hence, we discuss that building knowledgeable foundation models requires moving beyond the assumption that continuous weights can seamlessly

absorb discrete facts without explicit, mathematically grounded architectural mediation (Zhu et al., 2025; Pan et al., 2024).

## 2 The Anatomy of the Impedance Mismatch

To understand why simple text-based retrieval fails to achieve true semantic fusion, we must establish the differences between symbolic graphs and continuous vector spaces. The core technical challenge of integration lies in reconciling the continuous, statistical nature of neural networks with the discrete, logical nature of symbolic systems (d’Avila Garcez et al., 2019; Ji et al., 2022). We categorize this impedance mismatch across three structural dimensions: relational architecture, logical certainty, and memory editability.

### 2.1 Formalizing the Impedance Mismatch

To ground the impedance mismatch, we must formalize the structural degradation that occurs when mapping discrete relational architectures into continuous latent spaces (Bian, 2025).

Let a Knowledge Graph be defined as a discrete topological space  $\mathcal{K} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V}$  represents the set of entity vertices and  $\mathcal{E}$  represents the set of relational edges. This space is equipped with a shortest-path metric  $d_{\mathcal{K}}(v_i, v_j)$  that calculates the discrete logical distance between two entities  $v_i, v_j \in \mathcal{V}$ . Conversely, let the Foundation Model’s latent space be a continuous metric space  $\mathcal{M} \subseteq \mathbb{R}^h$ , where  $h$  denotes the dimensionality of the dense vectors, equipped with a geometric distance function  $d_{\mathcal{M}}$ . Any integration strategy requires a representation mapping function  $f : \mathcal{V} \rightarrow \mathcal{M}$ .

According to the principles of metric embedding theory, mapping an arbitrary discrete graph into a continuous vector space guarantees a strictly positive structural distortion. We formally define the Impedance Mismatch, denoted as  $\mathcal{I}$ , as the unavoidable mathematical lower bound of this distortion:

$$\mathcal{I} = \inf_f \left( \sup_{u \neq v} \frac{d_{\mathcal{M}}(f(u), f(v))}{d_{\mathcal{K}}(u, v)} \times \sup_{u \neq v} \frac{d_{\mathcal{K}}(u, v)}{d_{\mathcal{M}}(f(u), f(v))} \right)$$

where  $\inf_f$  denotes the infimum (greatest lower bound) over all possible mapping functions  $f$ , and  $\sup_{u \neq v}$  denotes the supremum (least upper bound) over all distinct pairs of entities  $u, v \in \mathcal{V}$ . In a

purely discrete, deterministic system,  $\mathcal{I} = 1$ , representing perfect structural isometry. However, for dense transformer representations,  $\mathcal{I} \gg 1$ . This formula shows that continuous spaces cannot faithfully preserve complex graph motifs, such as closed cycles and hierarchical trees, without warping the distances between nodes (Jin et al., 2024). Furthermore, this mismatch manifests as a compounding error during relational composition. In a discrete graph, navigating from a source entity  $v_1$  to a target entity  $v_3$  via sequential relations  $r_1$  and  $r_2$  is a deterministic algebraic composition, yielding an exact target node. In a foundation model, this multi-hop relation is approximated geometrically via sequential self-attention blocks. If  $A^{(l)}$  represents the attention matrix at layer  $l$ , and  $L$  represents the total number of attention layers, the continuous approximation introduces an error term  $\epsilon$ :

$$\epsilon = \left\| f(v_3) - \prod_{l=1}^L A^{(l)} f(v_1) \right\|$$

As the number of logical hops increases, the continuous approximation error  $\epsilon$  compounds multiplicatively. This formalizes exactly why text-based retrieval frameworks fail at multi-hop logical reasoning (Luo et al., 2025b; Kim et al., 2025): the continuous representation natively lacks the closed algebraic properties required to keep  $\epsilon$  at zero.

## 2.2 Structural versus Geometric Relations

In a knowledge graph, knowledge is defined structurally. A relation between a subject entity  $v_s$  and an object entity  $v_o$  via a predicate  $r$  is represented as an explicit, discrete edge  $(v_s, r, v_o) \in \mathcal{E}$ , where  $\mathcal{E}$  is the set of all edges in the graph (Hogan et al., 2021). Retrieving a fact or executing a multi-hop logical query relies on exact graph traversal. The expressive power of such representations depends heavily on the discrete structural motifs used to capture interactions.

Conversely, Foundation Models operate in continuous, high-dimensional vector spaces where internal states are represented by dense tensors (Brown et al., 2020; Touvron et al., 2023). Relations are not explicit edges but are instead approximated geometrically through implicit affine transformations and attention-weighted sums. While a knowledge graph queries adjacency via an indicator function or boolean matrix multiplication, a transformer layer models a relation by computing a soft

self-attention distribution (Vaswani et al., 2017):

$$\text{Attn}(Q, K, V) = \text{softmax} \left( \frac{QK^\top}{\sqrt{d_k}} \right) V$$

In this geometric space, the relational edge between two concepts is a dense similarity scalar in the attention matrix. This continuous perception struggles to preserve the strict structural constraints required for reliable, multi-step symbolic reasoning (Pan et al., 2024; Jin et al., 2024). When discrete graph architecture is forced into this continuous geometry, the crisp boundaries of symbolic motifs inevitably blur. This geometric blurring directly leads to hallucinated edges, invalid logical hops, and a degradation of verifiable inference (Luo et al., 2025b,a; Edge et al., 2025).

## 2.3 Certainty versus Probability

The second dimension of the mismatch concerns the truth representation of the encoded knowledge. Knowledge graphs are explicitly built on deterministic logic. An edge either exists or it does not, providing definitive, discrete representations of facts. This structural rigidity makes them suitable for precise querying and explainable, rule-based reasoning (Hogan et al., 2021; Ji et al., 2022).

However, foundational models are fundamentally probabilistic engines trained to minimize cross-entropy loss over token distributions to learn statistical regularities of language (OpenAI et al., 2024). Their internal representation of a fact is inherently statistical and highly contextual. Real-world knowledge is thus modeled not as a binary truth but as a continuous probability density. Merging these two paradigms can cause a structural collapse (Pan et al., 2024). Either the definitive certainty of the knowledge graph must be relaxed into a probabilistic embedding, which mathematically destroys its logical guarantees, or the continuous parameter space of the foundational model must be artificially thresholded to accommodate discrete rules (Luo et al., 2025a; Zhang, 2025). Standard hybrid predictors often assume conditional independence between extracted symbols to bridge this gap. Unfortunately, this assumption limits their ability to model complex interactions and leads to overconfident, miscalibrated predictions (Jin et al., 2024; Luo et al., 2025b).

## 2.4 The Editability Conflict

Another problem with this impedance mismatch is the difference in how the two systems update

their information. Knowledge graphs are highly dynamic and editable. Updating a fact or correcting an outdated relationship requires a straightforward  $O(1)$  operation, executing the direct insertion or deletion of a discrete edge  $(v_s, r, v_o)$  (Hogan et al., 2021).

Updating the parametric memory of a foundational model presents a very different theoretical challenge (De Cao et al., 2021; Mitchell et al., 2022). Knowledge in a transformer is heavily interconnected across multiple layers and attention heads via dense vector addition. Modifying a specific fact requires gradient descent or surgical weight perturbations, operations that are inherently unstable for lifelong editing (Meng et al., 2022; Yao et al., 2023). Recent studies in continuous knowledge editing reveal a significant performance decline in both knowledge update efficacy and retention as the number of sequential edits increases (De Cao et al., 2021; Hase et al., 2023). Because the representations are continuous and overlapping, altering the parameters to update one fact often causes degraded interference with adjacent, structurally unrelated knowledge (Meng et al., 2022; Yao et al., 2023; Mitchell et al., 2022). While novel techniques that disentangle and sparsify knowledge representations show promise in alleviating this decline, the fundamental editability conflict remains an unsolved barrier (Pan et al., 2024; Luo et al., 2025a). The distributed nature of the embedding space inherently resists the localized, surgical updates that discrete knowledge graphs effortlessly support.

### 3 A Hierarchy of Integration Strategies

To analyze neuro-symbolic research, we structure existing literature into a three-tiered maturity model. This hierarchy categorizes integration strategies based on how deeply the discrete knowledge graph penetrates the continuous architecture of the foundational model (Pan et al., 2024; Luo et al., 2025a; Jin et al., 2024). As summarized in Table 1, we can then isolate and expose the specific theoretical limitations inherent to each paradigm.

#### 3.1 Level 1: Lexical and Prompt Injection (Surface-Level)

The most common integration paradigm in industrial and academic settings operates entirely at the surface level. This is mostly realized through Knowledge Graph-Augmented Generation frame-

works (Lewis et al., 2020; Gao et al., 2024; Xu et al., 2024; Liu et al., 2025b). In this approach, an external retriever isolates a structurally relevant subgraph, serializes the discrete triples into natural language text, and concatenates this verbalized payload directly into the context window of the foundational model (Lewis et al., 2020; Chen et al., 2025). Recent frameworks have attempted to optimize by retrieving hypothetical reasoning paths to improve evidence selection or by deploying adaptive multi-hop algorithms to reduce the overall token payload (Edge et al., 2025; Liu et al., 2025b).

**Critique:** While this methodology is accessible and deployable, lexical injection functions as a superficial patch. It inherently suffers from inference latency and remains bottlenecked by context window limitations. Surface-level integration is susceptible to knowledge conflicts, where the model’s parametric memory overrides the retrieved context (Luo et al., 2025b; Pan et al., 2024). When the verbalized graph information logically contradicts the pre-trained continuous weights of the foundation model, the architecture frequently discards the prompt in favor of its statistical prior (Mallen et al., 2023; Wang et al., 2025; Luo et al., 2025b). Furthermore, serializing a complex multidimensional graph structure into a flat, linear token stream dismantles the structural motifs required for multi-hop logical deduction (Edge et al., 2025; Bian, 2025).

To formally demonstrate this limitation, we define the mathematical boundary of the Lexical Bottleneck. Let a knowledge subgraph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  possess an average branching factor  $b$  and require a logical reasoning depth of  $k$ . Let  $\mathcal{T}$  represent the token space of a foundational model with a maximum context window length  $L$ . Assuming a uniform average branching factor  $b$ , the number of distinct reasoning paths of length  $k$  diverging from a source entity is  $b^k$ . The total number of elements required to fully represent this reasoning subgraph scales geometrically as  $\mathcal{O}(b^k)$ .

If  $c \geq 1$  is the minimum number of tokens required to serialize a single graph element, the minimum token length to represent the subgraph is  $c \cdot \mathcal{O}(b^k)$ . By the Pigeonhole Principle, if this required length exceeds the fixed capacity  $L$ , any deterministic serialization function must truncate information. In classical logic, removing a single premise from a multi-hop chain invalidates the entire deductive path. Consequently, as the reasoning depth  $k$  scales, preserving the complete set of relational paths becomes mathematically impossible

without unbounded information loss.

### 3.2 Level 2: Representation Alignment (Embedding-Level)

To bypass the tokenization bottlenecks of text verbalization, the second tier of integration attempts to align the representations of the knowledge graph and the foundational model within a shared latent mathematical space. Methodologies typically employ Graph Neural Networks or sophisticated translation-based embedding techniques to encode the relational architecture of the discrete graph into dense continuous vectors (Bordes et al., 2013; Kipf and Welling, 2017; Jin et al., 2024; Yasunaga et al., 2021). These graph embeddings are then fused, concatenated, or aligned via multi-task contrastive learning objectives with the native text embeddings of the foundational model during an explicit forward pass or intermediate fine-tuning stage (Liu et al., 2025a; Luo et al., 2025a; Zhang, 2025).

Critique: Embedding-level alignment represents a significant step forward, yet it introduces a representational gap (Pan et al., 2024). Forcing a strict discrete graph into a continuous text embedding space necessitates a mathematical projection that degrades the strict relational properties of the original symbolic graph (Liu et al., 2025a; Bian, 2025). In this paradigm, the continuous vector space acts as a lossy compression algorithm for discrete logic. The system permanently loses the precise relational boundaries inherent to discrete symbols. Hence, while the foundational model gains broad domain awareness, it remains incapable of executing precise algorithmic graph traversals without hallucinating edges or conflating distinct semantic nodes (Luo et al., 2025b; Kiguchi et al., 2025).

To formalize this representational gap, we define the geometric boundary of Topological Collapse as a direct, bounded consequence of the Impedance Mismatch ( $\mathcal{I}$ ) established in Section 2.1. When mapping the discrete metric space of the graph  $\mathcal{K} = (\mathcal{V}, \mathcal{E})$  into the continuous latent space  $\mathcal{M}$  via an embedding function  $f$ , the structural distortion cannot be arbitrarily minimized.

According to Bourgain’s Embedding Theorem, embedding a finite metric space of  $|\mathcal{V}|$  points into a Euclidean space inherently introduces a minimum structural distortion mathematically bounded by  $\Omega(\log |\mathcal{V}|)$ . Therefore, we can formally bound the Impedance Mismatch for Level 2 integrations as  $\mathcal{I} \geq \Omega(\log |\mathcal{V}|)$ . As the size of the ontology grows, this minimum distortion grows logarithmi-

cally. Because a perfect, distance-preserving semantic alignment strictly requires  $\mathcal{I} = 1$ , achieving zero-distortion integration at the embedding level is mathematically impossible. The continuous vector space natively lacks the geometric capacity to preserve the discrete graph structure, unavoidably forcing distinct semantic nodes to overlap and destroying the boundaries required for precise algorithmic traversals.

### 3.3 Level 3: Architectural Integration (Attention-Level)

The most advanced frontier of current research involves directly modifying the internal computational routing of the transformer architecture to explicitly accommodate graph structures. Rather than treating the knowledge graph as an external text payload or an aligned input vector, these methodologies inject graph priors directly into the message-passing framework or the self-attention calculations of the model (Luo et al., 2025a; Yasunaga et al., 2021). Recent architectural innovations include Graph-Guided Attention modules that non-invasively rewire the native attention matrices of the foundational model based strictly on knowledge graph adjacency (Zhang, 2025; Zhai et al., 2026). Parallel frameworks utilize cross-attention mechanisms to inject semantic graph prompts dynamically across intermediate hidden layers (Hu et al., 2022).

Critique: While architecturally integrated models exhibit state-of-the-art empirical performance on complex reasoning benchmarks (Jin et al., 2024; Yasunaga et al., 2021), they remain theoretically incomplete. They are computationally expensive to scale. They still treat the knowledge graph as an externalized constraint that must be dynamically consulted rather than functioning as an internalized, native knowledge structure. The fundamental mathematical friction remains unresolved because the neural network is still relying on continuous attention weights to approximate discrete logical routing (Pan et al., 2024; Luo et al., 2025a). Until the underlying transformer architecture natively supports discontinuous structural subspaces within its residual stream, true semantic fusion will remain out of reach (Zhai et al., 2026).

To mathematically formalize this architectural limitation, we define the boundary of Attention Approximation Leakage. In a pure symbolic system, logical routing is executed via a discrete adjacency matrix  $A \in \{0, 1\}^{n \times n}$ . Architecturally integrated

foundational models attempt to approximate this discrete routing using continuous attention matrices  $A_{\text{soft}} \in (0, 1)^{n \times n}$ .

Because the standard attention mechanism relies on the softmax function, it strictly outputs positive probabilities. Approximating a hard, discrete zero (indicating no relationship) requires infinite negative logits, which is impossible in a stable training regime. Therefore, every non-adjacent node contributes a strictly positive residual leakage error  $\delta > 0$  during the message-passing calculation. When the model attempts to execute a multi-hop logical query of depth  $k$ , the routing calculation approximates  $(A_{\text{soft}})^k$ . As  $k$  increases, the continuous leakage error  $\delta$  compounds exponentially, leading to severe representation over-smoothing. The precise signal of the true discrete reasoning path is inevitably drowned out by the accumulated noise of the continuous space, proving that approximating discrete routing with continuous attention weights is mathematically unsustainable for deep logical deduction.

## 4 Core Bottlenecks Preventing True Fusion

To move past the design limits of current integration strategies and achieve true semantic fusion, the community must address three fundamental bottlenecks. These barriers represent incompatibilities between discrete structural constraints and continuous latent spaces.

### 4.1 Bottleneck A: The Curse of Differentiable Logic

A prevalent method for injecting discrete logic into continuous models utilizes differentiable logic frameworks, which relax Boolean connectives and quantifiers into continuous operators (Rocktäschel and Riedel, 2017; Evans and Grefenstette, 2018; van Krieken et al., 2022b). Soft relaxations algorithmically map strict truth values to the continuous interval  $[0, 1]$  via t-norms, s-norms, and fuzzy aggregation operators (van Krieken et al., 2022b; Manhaeve et al., 2018). However, this mapping introduces an optimization bottleneck. The resulting loss landscapes are non-linear and suffer from acute gradient saturation (Giunchiglia et al., 2022; Wang et al., 2019). Once a logical formula is nearly satisfied, the gradients vanish entirely, prematurely halting the optimization process before true semantic alignment is achieved (van Krieken et al., 2022b;

Minervini et al., 2019).

Furthermore, soft truth values break classical logical equivalences. In a discrete knowledge graph, De Morgan’s laws and contraposition hold absolute certainty. In a relaxed tensor space, these functionally equivalent symbolic rules often yield entirely divergent optimization paths (Giunchiglia et al., 2022; Wang et al., 2019). This inherent conflict makes robust constraint satisfaction mathematically unstable under stochastic gradient descent. Consequently, researchers are forced to choose between Boolean faithfulness and optimization amenability (van Krieken et al., 2022b; d’Avila Garcez et al., 2019).

### 4.2 Bottleneck B: Structural and Geometric Interference

The second barrier is structural and geometric interference. In a discrete graph, edges provide perfect relational insulation. Editing the relation between a subject node and an object node has no impact on adjacent graph edges. In a continuous representation space, such absolute geometric isolation is mathematically impossible (Meng et al., 2022; Elhage et al., 2021). When discrete symbolic structures are encoded into high-dimensional vectors, they overlap and blend within the same dense space (Elhage et al., 2021).

Updating parametric memory to modify a specific bound relation inherently warps the local geometry of the embedding representation space (Meng et al., 2022; Hase et al., 2023). As the number of overlapping facts in the residual stream increases, theoretical capacity limits are reached, and knowledge extraction operations inevitably suffer from catastrophic crosstalk (Yao et al., 2023; Zhong et al., 2024). Surgically editing a specific semantic relation can inadvertently alter adjacent, structurally unrelated knowledge (Meng et al., 2022; De Cao et al., 2021; Cohen et al., 2023). The fluid nature of the transformer’s residual stream lacks the strict orthogonality required to perfectly insulate discrete variables during continuous updates (Wang et al., 2024). This leads to the logical consistency breaking down entirely under minor parameter perturbations (Cohen et al., 2023; Zhong et al., 2024; Hase et al., 2023).

### 4.3 Bottleneck C: The Symbol Grounding Asymmetry

The final bottleneck centers on the asymmetry in symbol grounding (Harnad, 1990; Ji et al., 2022).

Integration Level	Mechanism	Formal Mathematical Bottleneck	Asymptotic Failure Mode
Level 1: Surface	Lexical Prompt Injection	Lexical Bottleneck: $\mathcal{O}(b^k) > L$	Context truncation; inability to encode exponential path complexity.
Level 2: Embedding	Latent Vector Alignment	Topological Collapse: $D(f) \geq \Omega(\log  \mathcal{V} )$	Semantic conflation; distortion of discrete relational boundaries.
Level 3: Architecture	Graph-Guided Attention	Approximation Leakage: Compounding softmax error $\delta$ in $(A_{\text{soft}})^k$	Representation over-smoothing; discrete signal drowned in continuous noise.

Table 1: A theoretical taxonomy of neuro-symbolic integration strategies, classified by their fundamental mathematical bottlenecks and asymptotic failure modes during multi-hop reasoning.

Knowledge graphs rely on unique entity identifiers to maintain strict referential integrity across diverse contexts (Hogan et al., 2021). On the other hand, foundational models process information through contextualized, distributed sub-word token representations (Brown et al., 2020; OpenAI et al., 2024).

Aligning abstract, immutable symbols with fluid data patterns remains a major theoretical challenge (Pan et al., 2024, 2023). While prior works attempt to bridge this gap using contrastive alignment or dedicated entity embeddings, these methods assume a static mapping that ignores the dynamically overlapping nature of language models (Pan et al., 2024; Luo et al., 2025a; Zhang, 2025). Natively integrating symbolic knowledge requires a mechanism to dynamically instantiate and bind discrete roles to continuous fillers without losing the strict identity of the original symbol (d’Avila Garcez et al., 2019; Smolensky, 1990). Until this structural asymmetry is mathematically resolved, hybrid models will continue to rely on shallow pattern matching rather than exhibiting true, provable compositional generalization (Lake et al., 2016; Bahdanau et al., 2019; Ruis et al., 2020).

## 5 A Roadmap for the Knowledge Lifecycle

To resolve the bottlenecks in Section 4 and the impedance mismatch, we build upon the framework established by (Dhayalkar, 2025b) to propose an actionable three-stage knowledge lifecycle roadmap that transcends lexical bridging.

### 5.1 Emergence (Pre-training): Structured Residual Streams

Current pre-training paradigms rely on unconstrained geometric optimization. This reliance directly causes the structural and geometric interference of factual knowledge observed during com-

plex reasoning tasks (Elhage et al., 2021; Bricken et al., 2023). However, recent breakthroughs in Representation Engineering demonstrate that high-level concepts naturally manifest as stable subspace directions or principal-eigenvector backbones within the transformer’s residual stream (Zou et al., 2025; Park et al., 2024). Furthermore, models can natively recover spatial separations that directly map to structured human concept categories (Wang et al., 2023; Li et al., 2023).

To formalize this phenomenon, we propose the architectural development of *Structured Residual Streams*. Rather than allowing facts to overlap arbitrarily across the entire embedding latent space, future architectures should introduce explicit graph-theoretic inductive biases during pre-training (Pan et al., 2024; Luo et al., 2025a). By applying regularization penalties that enforce orthogonal subspaces for distinct knowledge domains, discrete relational structures could emerge natively within the continuous weights. This would equip the model with an inherent, mathematically insulated structure, preventing the catastrophic crosstalk that currently degrades multi-hop reasoning (Fraday et al., 2020).

### 5.2 Injection (Inference): Latent Sub-graph Injection via VSAs

The industry standard of text-based retrieval is limited by tokenization bottlenecks and the high influence of the continuous parametric prior (Mallen et al., 2023; Lewis et al., 2020). To bypass this, we must shift from external lexical prompting to *Latent Sub-graph Injection*. We propose utilizing Vector Symbolic Architectures (VSAs) as the mathematical bridge to achieve this integration natively.

VSAs provide a well-defined algebraic framework using operations like binding, bundling, and permutation to represent complex discrete graph data within unified high-dimensional vector spaces (Kanerva, 2009; Kleyko et al., 2022). VSAs retain

fixed-dimensional vectors that align naturally with the native embeddings of the standard transformer architecture (Smolensky, 1990). By encoding a retrieved knowledge graph subgraph directly into a VSA hypervector, researchers can inject explicit role-filler bindings directly into the intermediate attention layers of the foundation model at inference time (Meng et al., 2022; Kanerva, 2009; Dhayalkar, 2025a). This bypasses the superficial text layer and forces the model to condition its generation on strict, mathematically bound relations rather than probabilistic text prompts.

### 5.3 Updating (Editing): Orthogonal Subspace Editing

The editability conflict requires a new mathematical approach to model updates. Current continuous knowledge editing regimes suffer from a performance decline in knowledge retention as sequential edits increase (Meng et al., 2022; Mitchell et al., 2022; De Cao et al., 2021). While recent methods have advanced the ability to update long-form knowledge using dynamic weight adjustments, they still grapple with coupling of the continuous vector space (Yao et al., 2023; Zhong et al., 2024).

To guarantee localized factual updates without neighborhood interference, we call for the formalization of *Orthogonal Subspace Editing*. Recent dissections of perturbation weights indicate that disentangled and sparsified knowledge representations can alleviate performance degradation during continuous editing (Hase et al., 2023). Building on this insight, we hypothesize that by projecting targeted factual edits strictly along orthogonal feature directions that do not activate unrelated semantic concepts, we can achieve updates that are mathematically equivalent to localized edge-insertion. This theoretical direction would allow foundational models to be patched dynamically and safely, finally bringing the reliable editability of symbolic knowledge bases to neural parameter spaces (Pan et al., 2024; Luo et al., 2025a; Meng et al., 2022).

## 6 Conclusion

Continuing to treat knowledge graphs merely as external databases or retrieval dictionaries fundamentally limits the evolutionary trajectory of foundation models. Throughout this paper, we have demonstrated that the current industrial standard of text-based retrieval acts only as a superficial patch over a much deeper structural divide. We

defined this divide as the Impedance Mismatch, a mathematical friction that occurs when attempting to force rigid, deterministic graph relational structures into fluid, probabilistic embedding spaces.

By categorizing existing integration attempts into a hierarchy of maturity, we revealed that neither lexical prompt injection nor continuous representation alignment can preserve the strict logical motifs required for reliable, multi-hop reasoning. The true barriers to semantic fusion are not engineering hurdles, but rather deep theoretical bottlenecks. The saturation of differentiable logic, the structural and geometric interference of continuous memory, and the fundamental asymmetry of symbol grounding collectively prevent standard transformer architectures from natively internalizing discrete symbolic structures.

To construct truly knowledgeable foundation models, the research community must move beyond the paradigm of lexical bridging. We must confront the fundamental mathematical friction between discrete certainty and continuous probability directly at the architectural level. By pursuing structured residual streams, latent sub-graph injection via vector-symbolic architectures, and orthogonal subspace editing, we can transition from models that mimic factual recall to systems that genuinely harbor structured, editable knowledge. Resolving this impedance mismatch is the necessary next step in the knowledge lifecycle, enabling a future where the precision of symbolic logic and the expressivity of parametric memory are seamlessly and mathematically fused.

## Limitations

While this paper establishes a rigorous mathematical foundation for neuro-symbolic integration, it focuses strictly on formal analysis and does not include empirical experiments. Consequently, our proposed frameworks currently serve as theoretical blueprints. Bridging these formalisms, such as Structured Residual Streams and VSA injection into scalable training regimes, represents a natural next step for empirical research. Additionally, because our models assume perfectly deterministic knowledge graphs, future work must explore how these strict geometric constraints adapt to the noise and contradictions inherent in real-world knowledge bases.

## References

- Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. 2019. [Systematic generalization: What is required and can it be learned?](#) *Preprint*, arXiv:1811.12889.
- Haonan Bian. 2025. [Llm-empowered knowledge graph construction: A survey.](#) *Preprint*, arXiv:2510.20345.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. [Translating embeddings for modeling multi-relational data.](#) In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.
- Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermy, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. [Towards monosemanticity: Decomposing language models with dictionary learning.](#) *Transformer Circuits Thread*. <https://transformer-circuits.pub/2023/monosemantic-features/index.html>.
- 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, and 12 others. 2020. [Language models are few-shot learners.](#) In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Jialin Chen, Houyu Zhang, Seongjun Yun, Alejandro Mottini, Rex Ying, Xiang Song, Vassilis N. Ioannidis, Zheng Li, and Qingjun Cui. 2025. [Gril: Knowledge graph retrieval-integrated learning with large language models.](#) *Preprint*, arXiv:2509.16502.
- Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2023. [Evaluating the ripple effects of knowledge editing in language models.](#) *Preprint*, arXiv:2307.12976.
- Artur d’Avila Garcez, Marco Gori, Luis C. Lamb, Luciano Serafini, Michael Spranger, and Son N. Tran. 2019. [Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning.](#) *Preprint*, arXiv:1905.06088.
- Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. [Editing factual knowledge in language models.](#) In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Sahil Rajesh Dhayalkar. 2025a. [Attention as binding: A vector-symbolic perspective on transformer reasoning.](#) *Preprint*, arXiv:2512.14709.
- Sahil Rajesh Dhayalkar. 2025b. [Neuro-symbolic reasoning: A roadmap of unsolved core questions.](#) *TechRxiv*, 2025(1210).
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitan, Robert Osazuwa Ness, and Jonathan Larson. 2025. [From local to global: A graph rag approach to query-focused summarization.](#) *Preprint*, arXiv:2404.16130.
- Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. [A mathematical framework for transformer circuits.](#) *Transformer Circuits Thread*. <https://transformer-circuits.pub/2021/framework/index.html>.
- Richard Evans and Edward Grefenstette. 2018. [Learning explanatory rules from noisy data.](#) *J. Artif. Int. Res.*, 61(1):1–64.
- E. Paxon Frady, Denis Kleyko, and Friedrich T. Sommer. 2020. [Variable binding for sparse distributed representations: Theory and applications.](#) *Preprint*, arXiv:2009.06734.
- 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.](#) *Preprint*, arXiv:2312.10997.
- Eleonora Giunchiglia, Mihaela Catalina Stoian, and Thomas Lukasiewicz. 2022. [Deep learning with logical constraints.](#) In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-2022*, page 5478–5485. International Joint Conferences on Artificial Intelligence Organization.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. [Realm: Retrieval-augmented language model pre-training.](#) *Preprint*, arXiv:2002.08909.
- Stevan Harnad. 1990. Harnad, s. (1990). the symbol grounding problem. *physica d: Nonlinear phenomena*, 42(1-3), 335-346. 42:335–346.
- Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandharioun. 2023. [Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models.](#) In *Thirty-seventh Conference on Neural Information Processing Systems*.

- Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. [Knowledge graphs](#). *ACM Computing Surveys*, 54(4):1–37.
- Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. [Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2240, Dublin, Ireland. Association for Computational Linguistics.
- Shaoyong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. [A survey on knowledge graphs: Representation, acquisition, and applications](#). *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514.
- Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2024. [Large language models on graphs: A comprehensive survey](#). *Preprint*, arXiv:2312.02783.
- Pentti Kanerva. 2009. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*, 1(2):139–159.
- Kanan Kiguchi, Yunhao Tu, Katsuhiko Ajito, Fady Alnajjar, and Kazuyuki Murase. 2025. [Multi-modal integration analysis of alzheimer's disease using large language models and knowledge graphs](#). *IEEE Access*, 13:113718–113735.
- Soohyeong Kim, Seok Jun Hwang, JungHyouon Kim, Jeonghyeon Park, and Yong Suk Choi. 2025. [Re-GraphRAG: Reorganizing fragmented knowledge graphs for multi-perspective retrieval-augmented generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5426–5443, Suzhou, China. Association for Computational Linguistics.
- Thomas N. Kipf and Max Welling. 2017. [Semi-supervised classification with graph convolutional networks](#). In *International Conference on Learning Representations*.
- Denis Kleyko, Mike Davies, Edward Paxon Frady, Pentti Kanerva, Spencer J. Kent, Bruno A. Olshausen, Evgeny Osipov, Jan M. Rabaey, Dmitri A. Rachkovskij, Abbas Rahimi, and Friedrich T. Sommer. 2022. [Vector symbolic architectures as a computing framework for emerging hardware](#). *Proceedings of the IEEE*, 110(10):1538–1571.
- Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2016. [Building machines that learn and think like people](#). *Preprint*, arXiv:1604.00289.
- 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. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. [Emergent world representations: Exploring a sequence model trained on a synthetic task](#). In *The Eleventh International Conference on Learning Representations*.
- Yu Liu, Yanan Cao, Xixun Lin, Yanmin Shang, Shi Wang, and Shirui Pan. 2025a. [Enhancing large language model for knowledge graph completion via structure-aware alignment-tuning](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20970–20984, Suzhou, China. Association for Computational Linguistics.
- Zhaotai Liu, Harald Sack, and Genet Asefa Gesese. 2025b. [Hyp-krag: Hypothetical path-based knowledge graph retrieval augmented generation with deepseek](#). In *RAGE-KG 2025: The Second International Workshop on Retrieval-Augmented Generation Enabled by Knowledge Graphs, co-located with ISWC 2025, November 2–6, 2025, Nara, Japan*, volume 4079 of *CEUR Workshop Proceedings*, pages 45 – 55. CEUR-WS.
- Linhao Luo, Carl Yang, Evgeny Kharlamov, and Shirui Pan. 2025a. [Integrating large language models and knowledge graphs for next-level agi](#). *Companion Proceedings of the ACM on Web Conference 2025*.
- Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Yuanfang Li, Chen Gong, and Shirui Pan. 2025b. [Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 41540–41565. PMLR.
- Chuangtao Ma, Yongrui Chen, Tianxing Wu, Arijit Khan, and Haofen Wang. 2025. [Large language models meet knowledge graphs for question answering: Synthesis and opportunities](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24578–24597, Suzhou, China. Association for Computational Linguistics.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. [When not to trust language models: Investigating effectiveness of parametric and non-parametric memories](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

- Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. 2018. [Deepproblog: Neural probabilistic logic programming](#). In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.
- Pasquale Minervini, Matko Bošnjak, Tim Rocktäschel, Sebastian Riedel, and Edward Grefenstette. 2019. [Differentiable reasoning on large knowledge bases and natural language](#). *Preprint*, arXiv:1912.10824.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022. [Fast model editing at scale](#). *Preprint*, arXiv:2110.11309.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, 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.
- Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeljanenko, Wen Zhang, Matteo Lissandrini, Russa Biswas, Gerard de Melo, Angela Bonifati, Edlira Vakaj, Mauro Dragoni, and Damien Graux. 2023. [Large language models and knowledge graphs: Opportunities and challenges](#). *Preprint*, arXiv:2308.06374.
- Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jipu Wang, and Xindong Wu. 2024. [Unifying large language models and knowledge graphs: A roadmap](#). *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.
- Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. [The linear representation hypothesis and the geometry of large language models](#). *Preprint*, arXiv:2311.03658.
- Tim Rocktäschel and Sebastian Riedel. 2017. [End-to-end differentiable proving](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M Lake. 2020. [A benchmark for systematic generalization in grounded language understanding](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 19861–19872. Curran Associates, Inc.
- Paul Smolensky. 1990. [Tensor product variable binding and the representation of symbolic structures in connectionist systems](#). *Artificial Intelligence*, 46(1):159–216.
- 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.
- Emile van Krieken, Erman Acar, and Frank van Harmelen. 2022a. [Analyzing differentiable fuzzy logic operators](#). *Artificial Intelligence*, 302:103602.
- Emile van Krieken, Erman Acar, and Frank van Harmelen. 2022b. [Analyzing differentiable fuzzy logic operators](#). *Artificial Intelligence*, 302:103602.
- 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.
- Po-Wei Wang, Priya L. Donti, Bryan Wilder, and Zico Kolter. 2019. [Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver](#). *Preprint*, arXiv:1905.12149.
- Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024. [Knowledge editing for large language models: A survey](#). *Preprint*, arXiv:2310.16218.
- Xi Wang, Procheta Sen, Ruizhe Li, and Emine Yilmaz. 2025. [Adaptive retrieval-augmented generation for conversational systems](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 491–503, Albuquerque, New Mexico. Association for Computational Linguistics.
- Zihao Wang, Lin Gui, Jeffrey Negrea, and Victor Veitch. 2023. [Concept algebra for \(score-based\) text-controlled generative models](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Ran Xu, Patrick Jiang, Linhao Luo, Cao Xiao, Adam Cross, Shirui Pan, Jimeng Sun, and Carl Yang. 2025. [A survey on unifying large language models and knowledge graphs for biomedicine and healthcare](#). In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '25)*, pages 6195–6205. PMID: 41858611; PMCID: PMC12995553.
- Zhentao Xu, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, and Zheng Li. 2024. [Retrieval-augmented generation with knowledge graphs for customer service question answering](#). In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024*, page 2905–2909. ACM.
- 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](#). In *Proceedings*

of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 10222–10240, Singapore. Association for Computational Linguistics.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. [QA-GNN: Reasoning with language models and knowledge graphs for question answering](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Songlin Zhai, Guilin Qi, Yue Wang, and Yuan Meng. 2026. [Knowledge fusion via bidirectional information aggregation](#). *Preprint*, arXiv:2507.08704.

Qinggang Zhang. 2025. [Enhancing large language models with reliable knowledge graphs](#). *Preprint*, arXiv:2506.13178.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. 2024. [Mquake: Assessing knowledge editing in language models via multi-hop questions](#). *Preprint*, arXiv:2305.14795.

Zihui Zhu, Yuqi Tang, Qiang Zhang, and Keyan Ding. 2025. [Synergizing large language models and knowledge graphs in science: A survey](#). In *NeurIPS 2025 AI for Science Workshop*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2025. [Representation engineering: A top-down approach to ai transparency](#). *Preprint*, arXiv:2310.01405.

# LLM-MemCluster: Empowering Large Language Models with Dynamic Memory for Text Clustering

Yuanjie Zhu<sup>1</sup> Liangwei Yang<sup>1\*</sup> Ke Xu<sup>1</sup> Weizhi Zhang<sup>1</sup>

Zihe Song<sup>1</sup> Jindong Wang<sup>2</sup> Philip S. Yu<sup>1</sup>

<sup>1</sup>University of Illinois Chicago

<sup>2</sup>William & Mary

{yzhu224, lyang84, kxu25, wzhan42, zsong29, psyu}@uic.edu

jwang80@wm.edu

## Abstract

Large Language Models (LLMs) are reshaping unsupervised learning by offering an unprecedented ability to perform text clustering based on their deep semantic understanding. However, their direct application is fundamentally limited by a lack of stateful memory for iterative refinement and the difficulty of managing cluster granularity. As a result, existing methods often rely on complex pipelines with external modules, sacrificing a truly end-to-end approach. We introduce **LLM-MemCluster**, a novel framework that reconceptualizes clustering as a fully LLM-native task. It leverages a **Dynamic Memory** to instill state awareness and a **Dual-Prompt Strategy** to enable the model to reason about and determine the number of clusters. Evaluated on several benchmark datasets, our tuning-free framework significantly and consistently outperforms strong baselines. LLM-MemCluster presents an effective, interpretable, and truly end-to-end paradigm for LLM-based text clustering.

## 1 Introduction

Text clustering, a cornerstone task in Natural Language Processing (NLP), aims to automatically organize a collection of documents into meaningful groups based on content similarity. This unsupervised learning technique is pivotal for large-scale knowledge discovery and information organization, with its utility demonstrated in applications ranging from structuring massive document archives to analyzing the collective voice of online communities (Zhou et al., 2024; Hadifar et al., 2019). Traditional clustering methods, such as K-Means (Jin and Han, 2017; Sinaga and Yang, 2020) or hierarchical clustering (Sahoo et al., 2006; Ran et al., 2023), typically operate on vector-space representations like TF-IDF (Bafna et al., 2016) or, more recently, pre-trained text embeddings from

benchmarks like MTEB (Muennighoff et al., 2023). While these approaches are effective, a notable limitation (Ezugwu et al., 2022) is their reliance on either handcrafted features or domain-specific fine-tuning to achieve optimal performance.

The advent of Large Language Models (LLMs) with powerful semantic understanding and reasoning capabilities, such as GPT-4, Gemini, and DeepSeek (Achiam et al., 2023; Team et al., 2023; Liu et al., 2024), has introduced a new paradigm for text clustering. Current research, however, has largely focused on hybrid frameworks that employ LLMs in auxiliary roles to enhance traditional embedding-based pipelines. These applications include enriching text representations (Wang et al., 2024), refining cluster assignments (Feng et al., 2024), and supervising the fine-tuning of external embedding models (Zhang et al., 2023). While innovative, these methods’ reliance on external components precludes fully LLM-native clustering.

However, using LLMs as standalone clustering agents reveals two fundamental architectural challenges. The first is a direct conflict between operational requirements and model design: the limited context window necessitates processing large datasets in batches, yet the models’ inherent statelessness prevents memory retention across these batches. This contradiction is a primary hurdle for achieving coherent and stable cluster assignments. This problem is further compounded by a second critical challenge: controlling clustering granularity. Without an explicit mechanism for guiding the partitioning process, LLMs tend to produce arbitrary and unstable topic partitions, as they lack an intrinsic method to determine a suitable degree of specificity. These limitations highlight the need for a framework that can impose statefulness while actively steering the clustering process.

To address these challenges, we introduce a novel framework for text clustering named **LLM-MemCluster**. This approach leverages large lan-

\*Corresponding author.

guage models, requires no model fine-tuning or integration with traditional algorithms, and is driven by two key innovations—each specifically designed to address the aforementioned limitations.

1. **Dynamic Memory Mechanism:** We introduce a memory mechanism that maintains a dynamic set of cluster labels within the prompt. This evolving memory state transforms the LLM into a state-aware clustering agent that can iteratively assign documents to existing clusters, create new ones for distinct topics, and merge and refine the cluster labels to ensure global consistency.
2. **Granularity Control Mechanism:** To actively guide the LLM in determining a suitable number of clusters, we employ two distinct prompting modes. A strict prompt encourages the consolidation of the existing cluster memory into broader categories, whereas a relaxed prompt fosters the discovery of more fine-grained topics. This dual-mode strategy enables the framework to explore different levels of granularity, ultimately achieving a stable and well-justified cluster count.

Our comprehensive experiments on several public benchmark datasets demonstrate that LLM-MemCluster significantly outperforms both traditional embedding-based methods and existing LLM-enhanced baselines across multiple standard evaluation metrics. These findings validate our framework as an effective solution for text clustering, harnessing the potential of end-to-end LLMs.

In summary, our contributions are threefold:

- Dynamic Memory Mechanism overcoming inherent LLM statelessness and facilitating the iterative refinement of cluster quality.
- Granularity Control Mechanism employing a novel dual-prompt strategy to ensure precise, user-guided control of cluster granularity.
- State-of-the-art performance on multiple standard clustering benchmarks, demonstrating robust, fine-tuning-free generalization across a diverse spectrum of both proprietary and open-source large language models.

## 2 Method

### 2.1 Problem Formulation

Text clustering aims to automatically organize a collection of documents into meaningful groups

based on content similarity. Formally, given an unlabeled text corpus,  $\mathcal{D} = \{x_1, x_2, \dots, x_N\}$ , the objective is to derive a partition of the corpus,  $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K\}$ . This partition consists of  $K$  clusters, where each cluster  $\mathcal{C}_k$  is a subset of the original corpus  $\mathcal{D}$ , formally defined as:

$$\mathcal{C}_k = \{x_j \in \mathcal{D} \mid l_j = k\}$$

Here,  $l_j$  represents the label assigned to instance  $x_j$ . The final partition covers all instances, with mutually exclusive clusters. The number of clusters,  $K$ , is dynamically determined, satisfying  $1 \leq K \leq N$ .

### 2.2 Framework Overview

We propose **LLM-MemCluster**, a novel framework that leverages API calls to a large language model (LLM), eliminating the need for model fine-tuning or integration with traditional algorithms. As illustrated in Figure 1, our framework is designed to directly address two principal challenges: the statelessness of LLMs and the inherent ambiguity in determining the number of clusters.

The architecture of LLM-MemCluster is centered on two synergistic innovations: a **Dynamic Memory** mechanism that endows the LLM with a functional state, and a **Dual-Prompt Strategy** for active control over clustering granularity. The framework processes each text instance from  $\mathcal{D}$  sequentially. Throughout this process, it maintains a dynamic set of assignments  $\mathcal{A} = \{(x_j, l_j)\}_{j=1}^N$ , recording the label  $l_j$  for each instance  $x_j$ . This set  $\mathcal{A}$  is crucial for the iterative refinement and is used to produce the final partition  $\mathcal{C}$ .

### 2.3 Stateful Clustering via Dynamic Memory

The inherent statelessness of contemporary LLMs, confining their operational memory to a single context window, presents a significant challenge for iterative tasks. In clustering, this leads to inconsistent assignments and redundant clusters. Our Dynamic Memory mechanism addresses this by providing the LLM with a persistent working memory of the evolving cluster landscape. Our framework operates in a single pass, efficiently completing the clustering of  $N$  instances in exactly  $N$  steps, unlike iterative methods like K-Means.

The memory module, denoted as  $\mathcal{M}_{\text{mem}}$ , maintains a dynamically updated set of descriptive labels representing the discovered clusters (e.g., “Arts”, “Science”). At each step  $i$  (for  $i = 1, \dots, N$ ), the framework processes instance  $x_i$ .

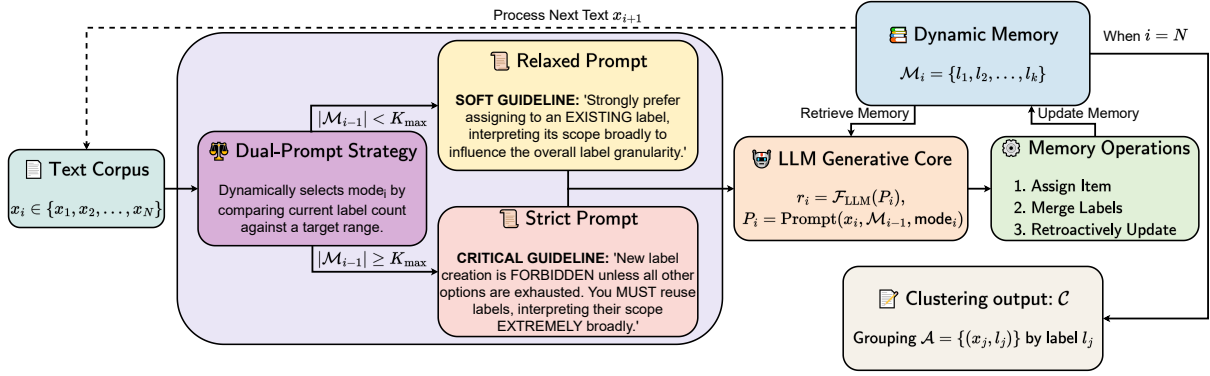


Figure 1: An overview of our proposed LLM-MemCluster framework. This figure illustrates the core iterative process, which is driven by a Dynamic Memory mechanism and the Dual-Prompt Strategy.

Let  $\mathcal{M}_{i-1}$  be the memory state before this step. The core operation is to invoke the LLM, modeled as a function  $\mathcal{F}_{\text{LLM}}$ , with a prompt constructed from the current instance  $x_i$ , the memory state  $\mathcal{M}_{i-1}$ , and the active mode (detailed in Section 2.4). Conceptually, the LLM returns a structured tuple containing an assignment label  $l_i$  and an optional merge suggestion  $s_i$ . A merge suggestion has the form  $(\mathcal{L}_{\text{old}}, l_{\text{new}})$ ; for example,  $s_i = (\{\text{“ML”}, \text{“DL”}\}, \text{“AI”})$  proposes consolidating existing labels. Formally:

$$(l_i, s_i) = \mathcal{F}_{\text{LLM}}(x_i, \mathcal{M}_{i-1}, \text{mode}_i). \quad (1)$$

The framework uses the returned label  $l_i$  and merge suggestion  $s_i$  to update its state, cycling through three core operations:

- **Reuse or Create.** For each text instance  $x_i$ , our framework constructs a prompt containing the current set of labels from  $\mathcal{M}_{i-1}$ . The LLM’s primary directive is to either reuse an existing label for  $x_i$  or create a new label if the text represents a fundamentally distinct topic, yielding the intermediate memory state  $\mathcal{M}'_i$ :

$$\mathcal{M}'_i = \begin{cases} \mathcal{M}_{i-1} \cup \{l_i\} & \text{if } l_i \text{ is a new label} \\ \mathcal{M}_{i-1} & \text{otherwise} \end{cases} \quad (2)$$

- **Merge and Refine.** A defining feature of our framework is its capacity to direct the LLM to propose a MERGE\_SUGGESTION at any step, enabling proactive consolidation of semantically similar or redundant labels. Crucially, this is an optional, concurrent action rather than a post-processing phase, allowing real-time optimization of the label space. The memory state  $\mathcal{M}_i$  is updated based on the merge suggestion  $s_i = (\mathcal{L}_{\text{old}}, l_{\text{new}})$ :

$$\mathcal{M}_i = \begin{cases} (\mathcal{M}'_i \setminus \mathcal{L}_{\text{old}}) \cup \{l_{\text{new}}\} & \text{on merge} \\ \mathcal{M}'_i & \text{otherwise} \end{cases} \quad (3)$$

- **Retroactive Update.** Upon receiving a merge suggestion, the framework updates  $\mathcal{M}_{\text{mem}}$  and retroactively re-maps historical assignments. This ensures global consistency (detailed in Appendix F). Specifically, any assignment  $(x_j, l_j) \in \mathcal{A}$  is updated to  $(x_j, l'_j)$ , where:

$$l'_j = \begin{cases} l_{\text{new}} & \text{if } l_j \in \mathcal{L}_{\text{old}} \\ l_j & \text{otherwise} \end{cases} \quad (4)$$

This integrated cycle transforms the stateless LLM into a state-aware clustering agent, ensuring both local accuracy and global consistency. The process is driven by a structured prompt (see Appendix A), which instructs the LLM to return a primary assignment—either reusing or creating a label—and, optionally, a merge suggestion.

## 2.4 Dual-Prompt Granularity Control

We introduce the Dual-Prompt Strategy to provide users with a means of actively guiding the final clustering granularity. This approach addresses the canonical challenge of steering the final number of clusters ( $K$ ) to align with user-defined goals, and is implemented as a dedicated control layer that actively regulates the cluster count. By doing so, the strategy ensures the final partition conforms to user expectations or the data’s intrinsic structure.

This strategy modulates the LLM’s propensity for new label creation by dynamically switching between two prompting modes. The mechanism is

guided by a user-defined target range for the cluster count,  $[K_{\min}, K_{\max}]$ . While the entire range is provided to the LLM as a contextual guideline for its decision-making, the programmatic switch between modes is triggered by the upper bound,  $K_{\max}$ . Specifically, the prompt mode for instance  $x_i$  is determined by the current cluster count:

$$\text{mode}_i = \begin{cases} \text{Strict} & \text{if } |\mathcal{M}_{i-1}| \geq K_{\max} \\ \text{Relaxed} & \text{otherwise} \end{cases} \quad (5)$$

This strategy uses two distinct prompt templates:

1. **The Strict Prompt:** Activated when the current number of clusters meets or exceeds the desired maximum, this mode incorporates prescriptive constraints into the prompt, significantly curtailing new label creation and compelling the LLM to prioritize **Reuse** and **Merge**. This raises the threshold for creating new clusters, hindering their formation.
2. **The Relaxed Prompt:** As the default operational mode, this prompt is used when the cluster count is within the desired range. It grants the LLM greater latitude in label creation, allowing it to form new clusters for semantically distinct topics as needed, thereby facilitating cluster discovery.

By adjusting prompt constraints based on the real-time cluster count, this strategy provides explicit control over the final clustering granularity, preventing uncontrolled label growth or premature consolidation. The complete strict and relaxed prompt templates are detailed in Appendix A. Finally, we provide the detailed algorithmic pseudocode and a computational complexity analysis in Appendix F.

### 3 Experiments

In this section, we evaluate our proposed framework, **LLM-MemCluster**, through experiments addressing the following research questions:

- **RQ1:** How does LLM-MemCluster perform against a variety of strong clustering baselines that employ different algorithms and state-of-the-art text representations?
- **RQ2:** How do the Dynamic Memory and Dual-Prompt Strategy components individually contribute to the overall effectiveness of our proposed LLM-MemCluster?

- **RQ3:** How robust is the LLM-MemCluster framework to variations in experimental conditions, specifically the dual-prompt transition threshold and the dataset execution order?
- **RQ4:** What is the generalization capability of the LLM-MemCluster framework when its foundational component is substituted with different large language models?

## 3.1 Experimental Setup

### 3.1.1 Datasets

We evaluate our method on six public benchmark datasets (Zhang et al., 2023), selected to cover a wide range of text clustering challenges. As detailed in Appendix B, these datasets span numerous domains and feature a broad range of cluster counts (K from 18 to 102), providing a robust testbed to assess the generalization of our method.

### 3.1.2 Evaluation Metrics

We evaluate performance using three standard metrics, where higher values indicate better performance and 1 denotes a perfect score:

- **Accuracy (ACC):** Calculates the percentage of correctly assigned data points, based on the optimal one-to-one mapping between predicted clusters and ground-truth labels.
- **Normalized Mutual Information (NMI):** Measures the mutual information between predicted and true labels, normalized by their entropies. It quantifies the statistical information shared between the two assignments.
- **Adjusted Rand Index (ARI):** A chance-adjusted measure of similarity between two data clusterings. It is calculated based on the proportion of sample pairs that are correctly assigned to the same or different clusters.

### 3.1.3 Baselines

We assess effectiveness by benchmarking against baselines from three distinct paradigms:

- **Traditional Method:** K-Means on TF-IDF vectors, a classic baseline relying on sparse, high-dimensional lexical features.
- **Embedding-based Methods:** We evaluate algorithms representing three key approaches: the centroid-based K-Means (Lloyd, 1982), the density-based DBSCAN (Deng, 2020),

and the graph-based Spectral Clustering (Ng et al., 2001). We apply these methods to instructor-large embeddings (Su et al., 2023) alongside the established BERTopic pipeline (Grootendorst, 2022).

- **LLM-based Method:** We compare against ClusterLLM (Zhang et al., 2023), which uses an LLM to generate pseudo-labels for training a smaller sentence encoder, enabling a highly scalable, multi-stage clustering approach. To ensure reproducibility, we set the temperature to 0 for all LLM-based experiments.

### 3.2 Main Results (RQ1)

As shown in Table 1, our framework, LLM-MemCluster, establishes a new state-of-the-art in unsupervised text clustering. On average, LLM-MemCluster surpasses the strongest baseline, ClusterLLM, by absolute margins of 9.5% in ACC, 4.4% in NMI, and 17.3% in ARI. The framework’s advantages are particularly evident on high-cardinality datasets where conventional methods tend to falter. For instance, on MTOP-I (K=102), it achieves an ARI of 68.9—a 38.9-point improvement over ClusterLLM. A similar 42.4-point gain in ARI on FewNerd (K=58) further demonstrates its effectiveness for semantically complex tasks.

These results offer a crucial insight: superior clustering performance is not merely a function of powerful text representations, but rather a result of an architectural design that effectively leverages these representations. This architectural dependence highlights the fundamental limitations of baseline methods. Embedding-based approaches, such as Spectral Clustering, rely on static vectors that, despite their quality, lack contextual adaptability. Other LLM-based methods like ClusterLLM treat the LLM as an external guide for knowledge distillation, rather than as a dynamic agent within the clustering process. This claim is further substantiated by our comprehensive generalization experiments in RQ4 (Section 3.5).

In contrast, the success of LLM-MemCluster is rooted in its novel architecture, which engages the LLM as a direct and active agent within a stateful, iterative process. The dynamic memory mechanism enables the framework to build a coherent, evolving understanding of the cluster space. This, in turn, allows the LLM to make adaptive, context-aware decisions at each step. We argue this direct and dynamic orchestration of the LLM’s

decision-making is the key innovation, allowing our method to navigate nuanced semantic relationships for more robust and accurate clustering.

### 3.3 Ablation Study (RQ2)

We conduct a comprehensive study to validate the contributions of our framework’s modules. The primary findings are summarized in Figure 2 and analyzed in detail in the subsequent subsections. Full numerical breakdowns for all experimental variants are provided in Appendix C.

**Memory and Grounding are Indispensable.** Figure 2a starkly highlights the critical roles of memory and in-context examples. Deactivating the Dynamic Memory (*w/o Memory*) causes a catastrophic performance degradation across all datasets, validating that an external memory is essential for overcoming LLM statelessness. The importance of grounding the model with few-shot examples is also evident, though its impact varies. Removing them (*w/o Few-shot*) generally leads to a significant ARI drop, a trend mirrored in Massive-D (from 53.8 to 44.0). However, the effect is exceptionally pronounced on FewNerd, where the ARI collapses from 53.1 to a mere 6.1. The stark performance drop on FewNerd underscores that for semantically complex domains, few-shot grounding is a prerequisite for robust performance.

**The Dual-Prompt Strategy is Highly Effective.** As shown in Figure 2b, the superiority of our dual-prompt approach is evident, as variants relying on a single prompt type consistently underperform the full model. This pattern is not only clear on average—where the full model achieves a 45.4 ARI, compared to 38.1 for the strict-only and 32.5 for the relaxed-only variants—but is also robustly replicated across individual datasets. For instance, on Massive-D the full model’s 53.8 ARI significantly exceeds the alternatives (43.1 and 37.0); a similar trend is observed on FewRel (32.7 vs. 26.5 and 20.8). This consistent underperformance validates our core design principle: a dynamic transition from an exploratory to a consolidative phase is the most effective strategy for reliably achieving optimal clustering granularity.

To further analyze how optimal granularity is achieved, Figures 2c to 2f illustrate the framework’s adaptive behavior on representative datasets. On Massive-I, the framework achieves a higher ARI score via **semantic splitting**, producing more clusters than the ground-truth by identifying fine-grained sub-topics. Conversely, on the high-

Method	ArxivS2S			Massive-I			MTOPI			Massive-D			FewNerd			FewRel			AVG		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
K-Means-TF-IDF	12.1	31.7	1.2	31.1	49.8	8.4	31.7	55.3	16.0	43.9	44.2	10.5	11.1	37.7	1.0	23.1	36.1	4.2	25.5	42.5	6.9
DBSCAN	6.3	17.6	0.3	20.4	28.9	1.0	21.5	26.6	2.2	25.9	30.7	6.4	27.0	0.5	0.1	10.5	19.5	0.4	18.6	20.6	1.7
Spectral	<u>25.1</u>	48.0	9.2	<b>60.5</b>	72.9	38.7	<u>39.2</u>	68.8	27.8	54.1	64.7	33.0	34.0	42.0	9.3	35.4	51.5	15.7	41.4	58.0	22.3
BERTopic	17.9	39.2	1.8	52.5	70.0	32.5	35.8	64.1	15.9	52.6	56.2	29.3	34.9	40.3	<u>11.7</u>	31.1	50.7	9.7	37.5	53.4	16.8
K-Means-Inst	<u>25.1</u>	49.3	12.3	<u>55.7</u>	72.6	41.6	34.5	70.9	26.9	<u>54.9</u>	<u>66.9</u>	<u>42.7</u>	28.2	43.3	6.1	34.8	53.1	22.5	38.9	59.3	25.4
ClusterLLM	<u>25.1</u>	<u>50.5</u>	<u>13.7</u>	55.5	<b>74.6</b>	<u>43.2</u>	36.0	<u>73.4</u>	<u>30.0</u>	52.4	65.3	40.8	<u>37.3</u>	<u>53.1</u>	10.7	<b>43.8</b>	<u>59.6</u>	<u>30.4</u>	<u>41.7</u>	<u>62.8</u>	<u>28.1</u>
<b>Our Method</b>	<b>28.4</b>	<b>57.4</b>	<b>16.3</b>	54.8	<u>73.5</u>	<b>47.9</b>	<b>64.0</b>	<b>77.5</b>	<b>68.9</b>	<b>57.6</b>	<b>67.7</b>	<b>53.8</b>	<b>59.3</b>	<b>63.3</b>	<b>53.1</b>	<u>43.2</u>	<b>63.6</b>	<b>32.7</b>	<b>51.2</b>	<b>67.2</b>	<b>45.4</b>

Table 1: Comparison of LLM-MemCluster with baselines across six datasets using ACC, NMI, and ARI scores (%). The best and second-best results are highlighted in **bold** and underlined, respectively. Baselines utilize instructor-large embeddings (except K-Means-TF-IDF), while our method employs in-context learning and ClusterLLM uses it to provide guidance (both utilizing the GPT-4.1 mini model).

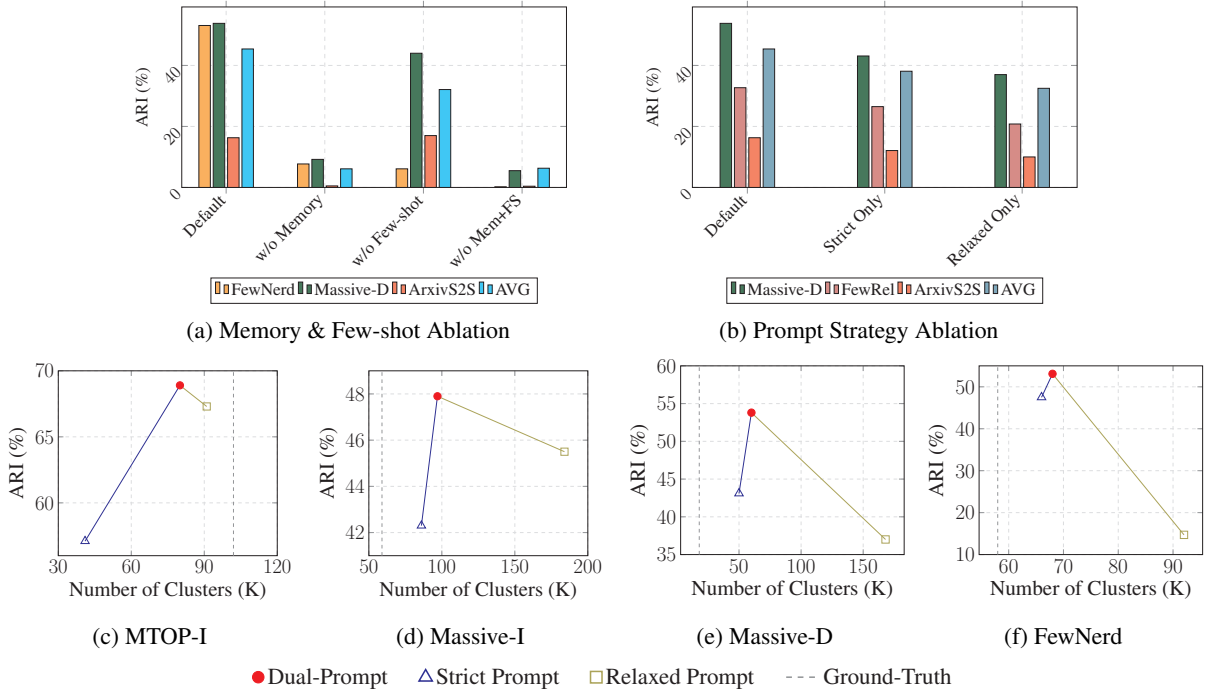


Figure 2: Comprehensive ablation study and adaptive clustering strategy comparison.

cardinality MTOPI dataset, it performs **semantic consolidation**, merging overly similar categories to produce fewer clusters. Crucially, in both scenarios, the Dual-Prompt strategy yields the solution with the highest ARI score. This demonstrates that the framework does not rigidly pursue a specific  $K$  but rather optimizes for semantic coherence, adaptively deciding whether to split or merge, a determination strictly contingent on the intrinsic semantic properties of each dataset.

### 3.4 Robustness Analysis (RQ3)

To address RQ3, we evaluate the robustness of LLM-MemCluster against variations in experimental conditions, specifically focusing on the hyperparameter sensitivity and the stability under randomized dataset execution orders.

**Hyperparameter Sensitivity** We first analyze the sensitivity to the core hyperparameter: the transition threshold for the Dual-Prompt Strategy. This threshold determines when the model switches from the initial, exploratory relaxed prompt to the subsequent, consolidative strict prompt. We operationalize this threshold as an offset applied to the upper bound of the target range ( $K_{max}$ ). A positive offset extends the exploratory phase, while a negative offset accelerates the consolidation process.

We evaluated a broad spectrum of offsets: -10, 0, +10, +50, +100, and +200. The results are visualized in Figure 3, which plots performance against different threshold offsets, with detailed results in Appendix D.1. While the average performance across all datasets (Figure 3c) shows rela-

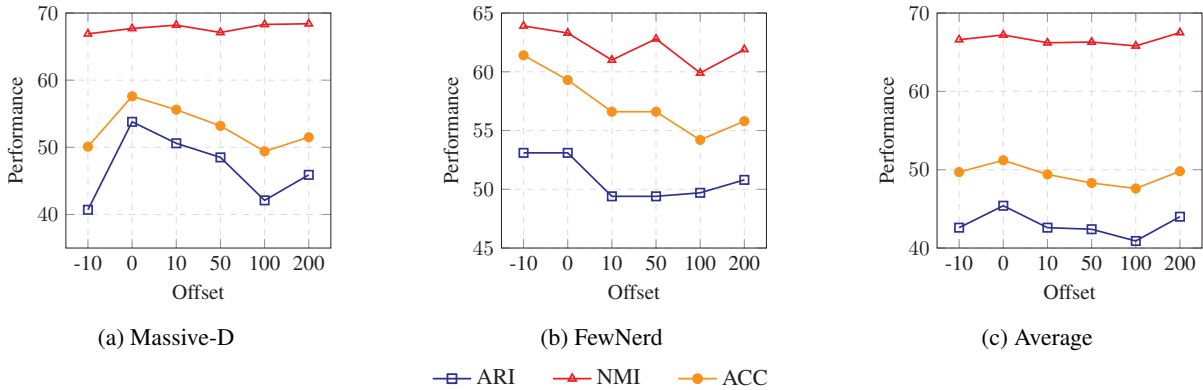


Figure 3: Hyperparameter sensitivity analysis of the prompt transition threshold, demonstrating robust and near-optimal performance across a wide range of values for representative datasets and on average.

tively flat curves, this stability is even more evident on individual datasets. For instance, on FewNerd (Figure 3b), the ARI score is exceptionally stable, fluctuating only minimally between a peak of 53.1 (default offset) and a low of 49.4. Even on Massive-D (Figure 3a), which exhibits more variance, performance peaks at an offset of 0 (53.8 ARI) and remains competitive across a wide range.

Notably, the framework performs well even at the extremes. An extended exploratory phase (offset +200) yields a strong ARI of 44.0 and the highest average NMI of 67.5. Conversely, an accelerated transition (offset -10) also maintains a robust 42.6 ARI. This resilience at the boundaries, mirrored across representative datasets, highlights the inherent robustness and self-correcting capacity of the dual-prompt mechanism, which adapts effectively to minor variations in the consolidation timing and control process.

**Seed Robustness** Beyond hyperparameter settings, we also examine the stability of our method under different dataset execution orders. This is a critical factor for stream clustering algorithms, where the processing sequence can inherently influence the resulting cluster integrity. We conducted experiments using 5 random seeds to simulate permuted input streams and report the mean and standard deviation in Appendix Table 7. The results demonstrate that LLM-MemCluster maintains consistent performance regardless of the input order. The standard deviation of the ARI scores remains low across all datasets, specifically falling within a tight 5% margin. For example, on Massive-I and MTOP-I, the standard deviations are merely 2.85% and 3.75%, respectively. This low variance statistically confirms that our dynamic memory and

adaptive dual-prompt mechanisms work in concert to resiliently capture global semantic structures independent of the local input sequence.

Consequently, our framework provides a practical advantage by delivering near-optimal, reproducible results across diverse datasets without requiring dataset-specific tuning or seed selection.

### 3.5 Generalization to Different LLMs (RQ4)

In addressing RQ4, we assess our framework’s generalization by evaluating it across a range of Large Language Models, including GPT-4.1-mini (default), GPT-3.5-turbo, GPT-4.1, Gemini-2.0-flash, Gemini-2.5-flash-preview-05-20, and DeepSeek-V3-0324, thereby confirming its portability.

The results, presented in Table 2, highlight the framework’s strong portability and robustness. Performance remains exceptionally strong when other high-capability models are used. For instance, substituting our default GPT-4.1-mini (45.4 ARI) with the more powerful GPT-4.1 yields a nearly identical ARI of 45.3. Similarly, competitive performance is observed with Gemini-2.5-flash-preview-05-20 (44.5 ARI). The strong results from other models, including DeepSeek-V3-0324 (38.7 ARI), confirm our design’s successful generalization across a diverse set of LLM backbones.

The advantages of our design are most apparent when paired with less capable models. For instance, our framework achieves a 34.6 ARI using Gemini-2.0-flash, significantly surpassing the 28.1 ARI of ClusterLLM, which uses the more capable GPT-4.1-mini (Tables 2 and 1). This comparison strongly indicates that our innovative design approach, rather than the underlying model’s intrinsic capability, primarily accounts for the gains.

Base LLM	ArxivS2S			Massive-I			MTOP-I			Massive-D			FewNerd			FewRel			AVG		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
<b>GPT-4.1-M</b>	28.4	57.4	16.3	54.8	73.5	47.9	64.0	77.5	68.9	57.6	67.7	53.8	59.3	63.3	53.1	43.2	63.6	32.7	51.2	67.2	45.4
GPT-3.5-T	35.9	59.5	19.8	43.5	70.0	38.2	52.6	75.0	54.0	55.0	65.4	45.1	42.0	62.9	24.8	25.2	49.9	14.6	42.4	63.8	32.8
GPT-4.1	29.8	60.4	18.6	64.0	77.3	54.6	67.8	80.9	69.3	52.8	69.5	47.3	59.1	73.4	51.1	48.4	69.4	30.8	53.6	71.8	45.3
Gemini-2.0-F	33.8	60.2	21.3	29.1	51.7	11.1	63.2	74.7	63.4	50.0	63.2	35.9	46.3	57.4	52.3	37.1	59.1	23.9	43.3	61.1	34.6
DeepSeek-V3	23.9	54.1	14.9	48.1	69.1	39.3	60.9	73.7	63.2	53.7	59.7	39.7	53.6	62.0	62.3	20.9	46.5	12.7	43.5	60.9	38.7
Gemini-2.5-F	34.8	68.4	24.9	50.6	77.3	41.6	60.3	80.4	60.5	54.5	67.4	43.8	65.7	76.4	59.4	48.3	70.4	36.9	52.4	73.4	44.5

Table 2: Generalization of the framework across large language models in ACC, NMI, and ARI (%). For brevity, we abbreviate model names: GPT-4.1-M (GPT-4.1-mini), GPT-3.5-T (GPT-3.5-turbo), Gemini-2.0-F (Gemini-2.0-flash), DeepSeek-V3 (DeepSeek-V3-0324), and Gemini-2.5-F (Gemini-2.5-flash-preview-05-20).

Beyond performance, we provide a detailed analysis of the monetary cost in Appendix E. A key efficiency advantage of our design is that token consumption is predominantly driven by the input context, whereas the output completion tokens remain minimal. Given that completion tokens often incur higher rates, this characteristic ensures that monetary costs remain manageable and predictable, even when scaling to premium models.

## 4 Related Work

### 4.1 LLM-Augmented Clustering

A prominent approach employs a Large Language Model (LLM) as a high-level “oracle” to augment or refine clustering pipelines that rely on external models. These methods distill the LLM’s semantic judgment to address specific, challenging parts of the clustering process. For instance, Cequel (Wang et al., 2025) generates pairwise constraints (Basu et al., 2004) to guide a downstream clustering algorithm. Other work focuses on refinement, where LLMEdgeRefine (Feng et al., 2024) re-assigns ambiguous “edge points” on the boundaries of initial clusters to enhance their integrity. A third approach, ClusterLLM (Zhang et al., 2023), leverages an LLM to generate supervisory signals from confusing document triplets to effectively fine-tune a smaller, more efficient sentence encoder. While pragmatic, these “LLM-as-oracle” frameworks are hybrid solutions and do not constitute an end-to-end generative clustering process.

### 4.2 End-to-End Generative Clustering

A more recent paradigm shift leverages LLMs as standalone clustering agents, bypassing traditional numerical algorithms. A representative approach within this paradigm reframes clustering as a classification problem. The T-CLC framework (Huang and He, 2025), for example, operates in two distinct

stages where it first prompts an LLM to generate candidate labels. However, this approach relies on a subset of ground-truth labels to seed the generation process, effectively shifting the task paradigm towards semi-supervised learning. In contrast, our work addresses the more challenging, fully unsupervised setting where no prior knowledge of the label space is available. Other related works have focused on improving the prompting process. ZeroDL (Jo et al., 2025), for instance, first performs an open-ended inference step to learn the dataset’s underlying distribution and then incorporates this meta-knowledge into a more data-aware prompt. However, this approach still treats clustering as a static inference task rather than a dynamic process with a continuously evolving state.

Our work, LLM-MemCluster, builds upon a generative paradigm but introduces a novel framework designed for efficient, iterative, single-pass clustering. In contrast to the multi-stage or static-inference approaches, it employs a **Dynamic Memory** to create a stateful process that continuously refines the cluster space (Xu et al., 2025; Liu et al., 2025). Furthermore, our **Dual-Prompt Strategy** provides an explicit mechanism for active granularity control throughout the process, addressing the challenge of dynamically determining the cluster count (Petnehazi and Aradi, 2025).

## 5 Conclusion

We introduce LLM-MemCluster, an end-to-end text clustering framework using a Dynamic Memory and Dual-Prompt Strategy to operate as a stateful, iterative agent, addressing the core challenges of LLM statelessness and cluster granularity. Its robust architecture achieves state-of-the-art performance, establishing an LLM-native paradigm that advances beyond hybrid approaches to effectively unlock LLM potential in unsupervised tasks.

## Limitations

Our framework entails specific trade-offs. First, it relies on the instruction-following capabilities of the LLM; while our approach is effective on advanced models, performance may vary on smaller architectures unable to strictly adhere to complex dual-prompt constraints. Second, as is inherent to its single-pass streaming nature, it optimizes based on the evolving memory state rather than performing multi-pass global refinement over the entire dataset. Finally, although the user-defined target range offers flexible granularity control, it necessitates minimal domain knowledge regarding the expected cluster distribution.

## Acknowledgments

This work is supported in part by NSF under grant III-2106758. This work used the Delta system at the National Center for Supercomputing Applications (award OAC 2005572) through allocation CIS251166 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Prafulla Bafna, Dhanya Pramod, and Anagha Vaidya. 2016. [Document clustering: Tf-idf approach](#). In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 61–66.
- Sugato Basu, Arindam Banerjee, and Raymond J Mooney. 2004. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 333–344. SIAM.
- Dingsheng Deng. 2020. Dbscan clustering algorithm based on density. In *2020 7th international forum on electrical engineering and automation (IFEAA)*, pages 949–953. IEEE.
- Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. [Few-NERD: A few-shot named entity recognition dataset](#). 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 3198–3213, Online. Association for Computational Linguistics.
- Absalom E Ezugwu, Abiodun M Ikotun, Olaide O Oyelade, Laith Abualigah, Jeffery O Agushaka, Christopher I Eke, and Andronicus A Akinyelu. 2022. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, 110:104743.
- Zijin Feng, Luyang Lin, Lingzhi Wang, Hong Cheng, and Kam-Fai Wong. 2024. [LLMEdgeRefine: Enhancing text clustering with LLM-based boundary point refinement](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18455–18462, Miami, Florida, USA. Association for Computational Linguistics.
- Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natara-jan. 2023. [MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4277–4302, Toronto, Canada. Association for Computational Linguistics.
- Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. [FewRel 2.0: Towards more challenging few-shot relation classification](#). 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 6250–6255, Hong Kong, China. Association for Computational Linguistics.
- Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.
- Amir Hadifar, Lucas Sterckx, Thomas Demeester, and Chris Develder. 2019. [A self-training approach for short text clustering](#). In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 194–199, Florence, Italy. Association for Computational Linguistics.
- Chen Huang and Guoxiu He. 2025. [Text clustering as classification with llms](#). In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, SIGIR-AP 2025*, page 374–384, New York, NY, USA. Association for Computing Machinery.

- Xin Jin and Jiawei Han. 2017. K-means clustering. In *Encyclopedia of machine learning and data mining*, pages 695–697. Springer.
- Hwiyeol Jo, Hyunwoo Lee, Kang Min Yoo, and Taiwoo Park. 2025. [ZeroDL: Zero-shot distribution learning for text clustering via large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19597–19607, Vienna, Austria. Association for Computational Linguistics.
- Haoran Li, Abhinav Arora, Shuohui Chen, Anshit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. [MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online. Association for Computational Linguistics.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Guangda Liu, Chengwei Li, Jieru Zhao, Chenqi Zhang, and Minyi Guo. 2025. [Clusterkv: Manipulating llm kv cache in semantic space for recallable compression](#). In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, pages 1–7.
- Stuart Lloyd. 1982. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.
- 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.
- Andrew Ng, Michael Jordan, and Yair Weiss. 2001. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14.
- Gabor Petnehazi and Bernadett Aradi. 2025. [Hercules: Hierarchical embedding-based recursive clustering using llms for efficient summarization](#). *arXiv preprint arXiv:2506.19992*.
- Xingcheng Ran, Yue Xi, Yonggang Lu, Xiangwen Wang, and Zhenyu Lu. 2023. Comprehensive survey on hierarchical clustering algorithms and the recent developments. *Artificial Intelligence Review*, 56(8):8219–8264.
- Nachiketa Sahoo, Jamie Callan, Ramayya Krishnan, George Duncan, and Rema Padman. 2006. Incremental hierarchical clustering of text documents. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 357–366.
- Kristina P Sinaga and Miin-Shen Yang. 2020. Unsupervised k-means clustering algorithm. *IEEE access*, 8:80716–80727.
- 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.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Hongtao Wang, Taiyan Zhang, Renchi Yang, and Jianliang Xu. 2025. [Cequel: Cost-effective querying of large language models for text clustering](#). In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management, CIKM '25*, page 2998–3008, New York, NY, USA. Association for Computing Machinery.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. [Improving text embeddings with large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916, Bangkok, Thailand. Association for Computational Linguistics.
- Wujiang Xu, Kai Mei, Hang Gao, Juntao Tan, Zujie Liang, and Yongfeng Zhang. 2025. [A-mem: Agentic memory for llm agents](#). *arXiv preprint arXiv:2502.12110*.
- Yuwei Zhang, Zihan Wang, and Jingbo Shang. 2023. [ClusterLLM: Large language models as a guide for text clustering](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13903–13920, Singapore. Association for Computational Linguistics.
- Sheng Zhou, Hongjia Xu, Zhuonan Zheng, Jiawei Chen, Zhao Li, Jiajun Bu, Jia Wu, Xin Wang, Wenwu Zhu, and Martin Ester. 2024. [A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions](#). *ACM Comput. Surv.*, 57(3).

## A Unified Prompt Template

This section details the unified prompt template at the core of our framework. As shown in Figure 4, the template integrates our **Dynamic Memory** by injecting the current Known labels, and our **Dual-Prompt Strategy** via placeholders for dynamic instructions. The specific content for the [SYSTEM\_GUIDELINE] and [USER\_CONSTRAINT] placeholders is provided in Figures 6 and 7.

```

--- SYSTEM PROMPT ---
You are an expert text analysis and clustering specialist.
Your primary goal is to determine the underlying theme, topic,
or relation type for each text input and assign it to an
appropriate category.

CORE PRINCIPLES:
- HIGHEST PRIORITY: Reuse existing labels whenever reasonably
  possible to ensure consistency.
- NEW LABELS: Create ONLY AS A LAST RESORT when an input is
  FUNDAMENTALLY NEW.
- MERGE: Suggest merging similar labels to improve conciseness.

[SYSTEM_GUIDELINE]

```

Figure 4: The unified prompt template (system prompt).

```

--- USER PROMPT ---
Known labels: ["label_1", "label_2", ...]

Examples:
Input: "Example text 1" -> Output: ASSIGNED_LABEL: "label_A"
Input: "Example text 2" -> Output: NEW_LABEL: "label_B"

Input to process: "text_to_cluster"

Instructions:
Your response must contain exactly one of the following primary lines:
- ASSIGNED_LABEL: <label_name>
- NEW_LABEL: <new_label_name> [USER_CONSTRAINT]

Optionally, you can also include the following line for consolidation:
- MERGE_SUGGESTION: MERGE: ["old_label"] INTO: ["new_label"]

RESPONSE FORMATTING:
- Exactly ONE 'ASSIGNED_LABEL:' OR 'NEW_LABEL:' line.
- Optionally, ONE 'MERGE_SUGGESTION:' line.

```

Figure 5: The unified prompt template (user prompt, with Known labels from dynamic memory).

```

[SYSTEM_GUIDELINE] Content
-----
# Relaxed Mode (Default)
SOFT GUIDELINE: As an additional consideration, try to manage the overall list of
known labels such that the total number of unique labels ideally stays {
range_desc}. This is a soft guideline to influence label granularity; your
primary decision-making process (prioritize reuse, create new only if essential,
suggest useful merges) remains paramount.

# Strict Mode
CRITICAL GUIDELINE: The total number of unique labels MUST be managed towards {
range_desc}. If approaching/exceeding the upper limit, new label creation is
SEVERELY RESTRICTED. You MUST aggressively reuse existing labels (interpret
their scope VERY broadly) and proactively seek merge opportunities.

```

Figure 6: Content for placeholder [SYSTEM\_GUIDELINE]. Injected into Figure 4 based on the mode.

## A.1 Dynamic Placeholder Content

Our framework modulates the prompt’s behavior by programmatically switching between two operational modes. Specifically, we dynamically populate two placeholders: [SYSTEM\_GUIDELINE]

(system-level guidance) and [USER\_CONSTRAINT] (user-specific constraints), as defined in Figure 4. The actual content injected into these placeholders consists of the detailed instructions and constraints shown in Figures 6 and 7. The transition between

```
[USER_CONSTRAINT] Content
-----
# Relaxed Mode
CONSIDERATION: If current known labels approach or exceed {target_max_clusters},
please be very cautious about creating NEW_LABEL. Strongly prefer assigning to
an EXISTING label (interpret its scope broadly) or identifying a MERGE.

# Strict Mode
CRITICAL CHECK: If current known labels approach or exceed {target_max_clusters},
creating a NEW_LABEL is FORBIDDEN unless all other options are exhausted. You
MUST first attempt to assign to an EXISTING label (interpret its scope EXTREMELY
broadly) or identify a MERGE. Only if the input is unequivocally unique and NO
existing label can accommodate it even with the broadest interpretation, and NO
merge is possible, then, as a final resort, create a NEW_LABEL.
```

Figure 7: Content for placeholder [USER\_CONSTRAINT]. Injected into Figure 4 based on the mode.

operational modes is governed by the number of discovered clusters relative to a user-defined upper bound,  $K_{\max}$ . The system operates in its **Relaxed Mode**, employing soft advisory language, as long as the cluster count remains below this threshold. Once  $K_{\max}$  is reached or exceeded, the system transitions to **Strict Mode**, using restrictive language to enforce the cluster cardinality.

## B Dataset Overview

Table 3 lists the datasets used in our experiments, detailing each one’s primary task or domain, the total number of samples, and the number of ground-truth clusters, denoted by  $K$ . We follow the experimental setup of Zhang et al. (2023) and use their processed versions of these datasets. The original sources are as follows: ArxivS2S is adapted from the MTEB benchmark (Muennighoff et al., 2023), consisting of scientific abstracts; Massive-I and Massive-D are subsets of the MASSIVE dataset (FitzGerald et al., 2023), focusing on intent detection and domain classification, respectively; MTOP-I is derived from the MTOP benchmark (Li et al., 2021), specifically focusing on intent classification; FewNerd is a large-scale, fine-grained named entity recognition dataset (Ding et al., 2021); and FewRel is a benchmark dataset for few-shot relation classification (Gao et al., 2019).

## C Detailed Ablation Study

### C.1 Analysis of Component Effectiveness

Tables 4 and 5 provide the detailed quantitative analysis comprehensively supplementing the ablation study discussion in Section 3.3.

**The Roles of Memory and Grounding** Our results in Table 4 clearly validate that both Dynamic

Dataset	Primary Task/Domain	# Samples	K
ArxivS2S	Scientific Abstracts	3,674	93
Massive-I	Intent Detection	2,974	59
MTOP-I	Intent Detection	4,386	102
Massive-D	Conversational Domain	2,974	18
FewNerd	Named Entity Recognition	3,789	58
FewRel	Relation Extraction	4,480	64

Table 3: Statistics of the datasets used in our experiments. K denotes the number of ground-truth clusters.

Memory and few-shot grounding are critical for the framework’s overall success.

- **Dynamic Memory (w/o Memory)**: Deactivating the memory module leads to a near-total collapse in performance across all six datasets. The average ARI consequently plummets from **45.4%** to a mere **6.1%**. This confirms that an external, stateful memory is essential to overcome the inherent statelessness of LLMs for iterative tasks like clustering.
- **Few-shot Grounding (w/o Few-shot)**: Removing the few-shot examples also causes a significant performance degradation, with the average ARI dropping from **45.4%** to **32.1%**. The effect is particularly dramatic on semantically nuanced datasets like FewNerd, where the ARI score collapses from **53.1%** to just **6.1%**. This highlights that for complex domains, providing in-context examples is crucial for effectively guiding the model to produce accurate and consistent outputs.

**Effectiveness of the Dual-Prompt Strategy** By conducting a cross-referenced analysis of the performance metrics in Table 4 and the generated clusters from Table 5, we can see how the Dual-Prompt

Method Variant	ArxivS2S			Massive-I			MTOP-I			Massive-D			FewNerd			FewRel			AVG		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
<b>Default</b>	<b>28.4</b>	<b>57.4</b>	<b>16.3</b>	<b>54.8</b>	<b>73.5</b>	<b>47.9</b>	<b>64.0</b>	<b>77.5</b>	<b>68.9</b>	<b>57.6</b>	<b>67.7</b>	<b>53.8</b>	<b>59.3</b>	<b>63.3</b>	<b>53.1</b>	<b>43.2</b>	<b>63.6</b>	<b>32.7</b>	<b>51.2</b>	<b>67.2</b>	<b>45.4</b>
w/o Memory	4.7	71.1	0.5	8.5	65.2	1.8	11.2	61.7	2.8	17.0	57.7	9.2	15.1	61.4	7.7	20.9	69.8	14.8	12.9	64.5	6.1
w/o Few-shot	28.1	58.5	17.0	50.2	70.9	43.5	61.4	74.3	60.5	55.5	65.7	44.0	27.3	48.7	6.1	33.5	55.3	21.2	42.7	62.2	32.1
w/o M+FS	4.2	71.0	0.4	18.5	67.4	11.9	23.1	65.2	18.5	11.7	55.4	5.5	4.3	57.2	0.2	5.3	66.2	1.2	11.2	63.7	6.3
Strict Prompt	18.6	49.0	12.1	52.6	69.7	42.3	56.8	74.8	57.1	50.2	67.6	43.1	57.8	62.6	47.5	38.1	64.1	26.5	45.7	64.6	38.1
Relaxed Prompt	17.5	60.6	10.0	54.7	74.3	45.5	65.4	78.3	67.3	41.7	65.4	37.0	39.4	52.1	14.7	33.7	56.6	20.8	42.1	64.5	32.5

Table 4: Ablation study of LLM-MemCluster supporting the analysis for RQ2. We report ACC, NMI, and ARI (%), highlighting the importance of each component by comparing performance to the Default setting.

Method Variant	ArxivS2S	Massive-I	MTOP-I	Massive-D	FewNerd	FewRel
Ground-Truth	<b>93</b>	<b>59</b>	<b>102</b>	<b>18</b>	<b>58</b>	<b>64</b>
Dual-Prompt	159	97	80	60	68	122
Strict Prompt	70	86	41	50	66	89
Relaxed Prompt	1208	184	91	168	92	141

Table 5: Comparison of the number of clusters (K) produced by different model variants.

strategy is superior to single-prompt variants.

- **Relaxed Prompt Variant:** This variant consistently generates a vastly larger number of clusters than the ground-truth (e.g., **1208** vs. 93 on ArxivS2S; **184** vs. 59 on Massive-I). This tendency to over-split the data results in poor semantic grouping and leads to the lowest average ARI of **32.5%**.
- **Strict Prompt Variant:** In contrast, the variant is overly conservative, often producing a cluster count that is suboptimal for that dataset (e.g., only **41** clusters for MTOP-I, where the ground-truth is 102). While this consolidation can be beneficial, it often merges distinct topics, capping its average ARI at **38.1%**.
- **Dual-Prompt Strategy:** The Dual-Prompt demonstrates a powerful adaptive capability. It navigates the trade-off between over-splitting and over-consolidating, producing a cluster count (e.g., **97** on Massive-I, **80** on MTOP-I) that better reflects the underlying data structure. This adaptive, dynamic control over granularity is the key reason it achieves the state-of-the-art average ARI of **45.4%**, outperforming both single-prompt baselines by a significant margin.

## D Detailed Robustness Analysis

### D.1 Hyperparameter Sensitivity

Table 6 details the full numerical results supporting the robustness claims in Section 3.4.

**Overall Performance Stability** The average performance across all datasets demonstrates remarkable stability. The average ARI remains high across a wide spectrum of offsets, from an accelerated transition (offset -10, ARI **42.6%**) to a significantly extended exploratory phase (offset +200, ARI **44.0%**). The peak performance is achieved at the default offset of 0 (ARI **45.4%**), but even extreme variations do not lead to a collapse in performance, underscoring the inherent self-correcting nature of the Dual-Prompt strategy.

**Dataset-Specific Robustness** The framework’s demonstrated robustness is not merely a statistical artifact of averaging; rather, it is consistently evident at the individual dataset level.

- On FewNerd, a semantically complex dataset, the ARI score proves to be exceptionally stable. It peaks at a high of **53.1%** (offsets 0 and -10), while its lowest point remains a robust **49.4%** (offset +10 and +50). This narrow range of fluctuation powerfully highlights the model’s ability to achieve consistent clustering results, regardless of potential minor timing adjustments in the consolidation phase.
- On Massive-D, which exhibits more variance, performance still remains competitive. While the peak ARI of **53.8%** is at the default offset, even an early transition (offset -10) yields a respectable ARI of **40.7%**, and a late transition (offset +200) maintains an ARI of **45.9%**.
- Notably, on some datasets like Massive-I, strategically shifting to an earlier transition (offset -10, ARI **52.4%**) or a later one (offset +200, ARI **52.3%**) can even outperform the default setting (ARI **47.9%**), suggesting that while the default is a strong general-purpose choice, the framework is robust enough to accommodate diverse data distributions.

Offset	ArxivS2S			Massive-I			MTOPI			Massive-D			FewNerd			FewRel			AVG		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
<b>0</b>	<b>28.4</b>	<b>57.4</b>	<b>16.3</b>	<b>54.8</b>	<b>73.5</b>	<b>47.9</b>	<b>64.0</b>	<b>77.5</b>	<b>68.9</b>	<b>57.6</b>	<b>67.7</b>	<b>53.8</b>	<b>59.3</b>	<b>63.3</b>	<b>53.1</b>	<b>43.2</b>	<b>63.6</b>	<b>32.7</b>	<b>51.2</b>	<b>67.2</b>	<b>45.4</b>
-10	27.4	57.0	15.9	58.2	72.9	52.4	59.7	75.5	63.4	50.1	66.9	40.7	61.4	63.9	53.1	41.6	63.7	30.5	49.7	66.6	42.6
+10	28.0	58.6	16.6	53.8	72.6	45.9	62.7	76.2	65.9	55.6	68.2	50.6	56.6	61.0	49.4	39.6	60.8	27.3	49.4	66.2	42.6
+50	25.0	56.7	14.2	55.0	74.6	48.8	61.2	76.1	66.2	53.2	67.1	48.5	56.6	62.8	49.4	39.1	60.8	27.0	48.3	66.3	42.4
+100	25.6	57.3	14.3	53.0	72.2	44.4	62.8	75.7	65.7	49.4	68.3	42.1	54.2	59.9	49.7	40.7	61.4	29.4	47.6	65.8	40.9
+200	27.7	61.5	16.7	57.9	74.4	52.3	63.7	76.9	68.2	51.5	68.4	45.9	55.8	61.9	50.8	42.0	62.2	30.5	49.8	67.5	44.0

Table 6: Hyperparameter analysis of the dual-prompt transition threshold. We report ACC, NMI, and ARI (%) across various switching offsets, with the default setting (offset 0) included for comparison.

Dataset	ACC	NMI	ARI
ArxivS2S	30.13±3.81	53.52±2.09	15.82±2.20
Massive-I	53.89±2.36	72.50±2.17	45.69±2.85
MTOPI	64.46±2.28	79.50±0.40	67.14±3.75
Massive-D	63.16±2.77	69.48±1.72	53.38±4.05
FewNerd	64.04±2.47	69.99±2.30	61.86±2.06
FewRel	30.81±4.52	54.92±5.06	19.34±4.17

Table 7: Robustness results using Gemini-2.5-Flash-Lite. Values represent mean ± std (%) computed over 5 random permutations of the dataset.

Table 6 confirms that LLM-MemCluster is not dependent on precise hyperparameter tuning. Its strong performance across a wide range of offsets enables near-optimal results on diverse datasets without laborious optimization.

## D.2 Seed Robustness Under Randomized Dataset Execution Order

Table 7 presents the stability of our framework under randomized dataset execution orders. We report the mean and standard deviation of ACC, NMI, and ARI across five random seeds using Gemini-2.5-Flash-Lite. These detailed statistics support the findings in Section 3.4, confirming that the framework achieves highly consistent performance independent of the specific input sequence.

## E Token Consumption and Cost

Table 8 details the token consumption and estimated monetary cost incurred by our framework using Gemini-2.5-Flash-Lite. We report the aggregated input and output tokens for each dataset, alongside the total cost derived from public pricing at the time of experimentation (\$0.10 per 1M input tokens and \$0.40 per 1M output tokens).

While tokenization and pricing vary across models, a consistent structural feature of our framework is that operational costs are predominantly driven

Dataset	Input Tokens	Output Tokens	Total Cost (USD)
ArxivS2S	4,495,870	32,969	0.46
Massive-I	3,590,409	25,027	0.37
MTOPI	4,537,051	33,653	0.47
Massive-D	3,231,711	31,226	0.34
FewNerd	4,821,283	33,393	0.50
FewRel	5,451,649	42,872	0.56
<b>Total</b>	<b>26,127,973</b>	<b>199,140</b>	<b>2.70</b>

Table 8: Token consumption and estimated cost for running LLM-MemCluster with Gemini-2.5-Flash-Lite on each dataset. Input token price is \$0.10 per 1M tokens and output token price is \$0.40 per 1M tokens.

by inexpensive input tokens, whereas the volume of costly output tokens remains minimal.

## F Algorithmic Implementation and Computational Complexity

The procedural implementation of our framework is detailed across two algorithms. Algorithm 1 describes the core, single-step clustering operation, which encapsulates the Dynamic Memory mechanism. Algorithm 2 then presents the main workflow of LLM-MemCluster, illustrating how the core operation and the Dual-Prompt Granularity Control are integrated to process the entire dataset.

LLM-MemCluster processes a corpus of  $N$  instances in a single, deterministic pass. For each instance, the primary computational costs stem from the LLM API call,  $C_{LLM}$ , and a potential retroactive update. A retroactive update incurs a cost of  $O(i)$  at step  $i$ . While the theoretical worst-case complexity is  $O(N^2)$ , empirical evidence consistently shows that merge events are remarkably rare (averaging fewer than 2 per dataset in our experiments with Gemini-2.5-Flash-Lite). Consequently, the effective complexity remains near-linear,  $O(N \cdot (C_{LLM} + C_{update}))$ , avoiding the non-deterministic convergence behavior inher-

---

**Algorithm 1** Core Clustering Operation

---

**Input:** Text instance  $x_i$ ; Memory of labels  $\mathcal{M}_{\text{mem}}$ ; Assignments  $\mathcal{A}$ ; Prompting *mode* (Relaxed/Strict)

**Output:** Updated  $(\mathcal{M}_{\text{mem}}, \mathcal{A})$

```
1:  $\mathcal{L}_{\text{seen}} \leftarrow \mathcal{M}_{\text{mem}}$ 
2:  $(l_i, s_i) \leftarrow \mathcal{F}_{\text{LLM}}(x_i, \mathcal{L}_{\text{seen}}, \text{mode})$       Eq. (1)
3:  $\mathcal{A} \leftarrow \mathcal{A} \cup \{(x_i, l_i)\}$ 
4: if  $l_i \notin \mathcal{L}_{\text{seen}}$  then
5:   Add  $l_i$  to memory  $\mathcal{M}_{\text{mem}}$       Eq. (2)
6: end if
7: if  $s_i$  is not null then
8:    $\mathcal{L}_{\text{old}}, l_{\text{new}} \leftarrow$  Extract labels from  $s_i$ 
9:    $\mathcal{M}_{\text{mem}} \leftarrow (\mathcal{M}_{\text{mem}} \setminus \mathcal{L}_{\text{old}}) \cup \{l_{\text{new}}\}$   Eq. (3)
10:  for each  $(x_j, l_j) \in \mathcal{A}$  do
11:    if  $l_j \in \mathcal{L}_{\text{old}}$  then
12:       $l_j \leftarrow l_{\text{new}}$       Eq. (4)
13:    end if
14:  end for
15: end if
```

---

---

**Algorithm 2** LLM-MemCluster Workflow

---

**Input:** Unlabeled text corpus  $\mathcal{D} = \{x_1, \dots, x_N\}$ ; LLM  $\mathcal{F}_{\text{LLM}}$ ; Target K range  $[K_{\text{min}}, K_{\text{max}}]$

**Output:** A partition of the corpus,  $\mathcal{C}$ .

```
1: Initialize memory  $\mathcal{M}_{\text{mem}} \leftarrow \emptyset$  and assignments  $\mathcal{A} \leftarrow \emptyset$ 
2: CoreOp updates  $\mathcal{M}_{\text{mem}}$  and  $\mathcal{A}$  in-place.
3: for each text instance  $x_i$  in  $\mathcal{D}$  do
4:   if  $|\mathcal{M}_{\text{mem}}| \geq K_{\text{max}}$  then
5:      $\text{mode} \leftarrow$  Strict
6:   else
7:      $\text{mode} \leftarrow$  Relaxed
8:   end if
9:    $\text{CoreOp}(x_i, \mathcal{M}_{\text{mem}}, \mathcal{A}, \text{mode})$ 
10: end for
11: Generate the final partition  $\mathcal{C}$  by grouping all instances in  $\mathcal{A}$  by their assigned label
12: return Final partition  $\mathcal{C}$ 
```

---

ent to iterative algorithms like K-Means.

In contrast to contemporary methods like ClusterLLM, which employs a multi-stage pipeline to fine-tune a separate encoder, our framework is a unified, single-pass procedure. This design avoids the costly overhead of intermediate model training and multiple algorithmic phases.

# Test-Time Training for Zero-Resource Dense Retrieval Reranking

Shiyan Liu<sup>1</sup> Yichen Li<sup>2</sup>

<sup>1</sup>Huazhong University of Science and Technology   <sup>2</sup>ByteDance  
shyl@hust.edu.cn   liyichen.1@bytedance.com

## Abstract

Dense retrievers excel at first-stage candidate generation but lack effective reranking in zero-resource settings. Existing approaches face a fundamental dilemma: cross-encoders deliver strong reranking quality but require costly supervised training and incur high latency, while unsupervised BM25 reranking consistently degrades dense retrieval performance on most of BEIR benchmarks. We propose **DART** (Dense Adaptive Reranking at Test-time), which resolves this dilemma by adapting the scoring function at inference time. For each query, the top-ranked documents serve as pseudo-positive examples and the bottom-ranked as pseudo-negative examples, providing noisy but readily available supervision to adapt a bilinear scoring matrix  $W$  via a small number of gradient updates. We further introduce a confidence-weighted margin loss and a cross-query momentum buffer that warm-starts adaptation across queries. On six BEIR benchmarks, DART achieves a mean per-dataset relative NDCG@10 gain of **+2.1%** over the dense retrieval baseline with under 10ms additional latency per query, demonstrating a powerful capability for zero-shot performance enhancement and cross-domain generalization.

## 1 Introduction

The modern information retrieval pipeline is typically organized as a two-stage cascade: a fast first-stage retriever narrows the corpus to a candidate set, which is then reranked by a more precise but computationally expensive model (Lin et al., 2022; Guo et al., 2020). Bi-encoder dense retrievers (Karpukhin et al., 2020; Reimers and Gurevych, 2019) have become the standard first stage, offering strong recall with sub-millisecond per-document scoring. However, reranking remains an open problem in *zero-resource* deployments where no labeled relevance judgments exist for the target domain.

Supervised rerankers address this with extra training. Cross-encoders (Nogueira and Cho, 2019;

Nogueira et al., 2020) jointly attend to query and document, achieving high accuracy at the cost of 200–500ms latency and substantial labeled data. Recent LLM-based rerankers (Sun et al., 2023; Weller et al., 2025) push accuracy further but amplify both requirements. In the absence of training data, practitioners typically fall back to the dense retrieval ranking itself—forgoing any reranking step entirely—because no lightweight, reliable alternative exists. This is especially true in deployments built entirely around vector databases (Johnson et al., 2019), where only dense embeddings are indexed and lexical systems such as BM25 (Robertson and Zaragoza, 2009) are not available.

We observe that a useful supervision signal is already present at inference time, without any external resource: the ranked list produced by the dense retriever itself. The top-ranked documents for a given query are likely relevant; the bottom-ranked are likely not. Although this pseudo-labeling is noisy, it captures query-specific relevance structure that the fixed, query-agnostic cosine scoring function cannot exploit. This motivates a *Test-Time Training* (TTT) approach (Sun et al., 2020; Liu et al., 2021): rather than modifying query or document representations, we adapt the *scoring function* directly for each incoming query using only its own retrieved candidates as supervision.

We propose **DART**, which frames reranking as a per-query optimization problem. Given a query, we initialize a bilinear scoring matrix  $W$  to the identity and perform a small number of gradient steps using a confidence-weighted margin loss over pseudo-labeled positives and negatives drawn from the top- $K$  retrieved documents. We additionally introduce a cross-query momentum buffer that accumulates adaptation signals across the query stream to warm-start each new query, and a dataset-adaptive optimizer selection strategy that balances convergence speed against pseudo-label noise. Evaluated on six BEIR benchmarks (Thakur et al., 2021), DART

achieves a mean per-dataset relative NDCG@10 gain of +2.1% over the dense retrieval baseline with under 10ms latency per query.

Our contributions are summarized as follows:

- We propose DART, a principled TTT framework for zero-resource dense retrieval reranking that adapts a bilinear scoring matrix at inference time using pseudo-labels derived directly from the dense retrieval ranking, requiring no external resource.
- We empirically demonstrate that DART improves over the dense retrieval baseline on five of six BEIR benchmarks with a mean per-dataset relative NDCG@10 gain of +2.1% and under 10ms additional latency per query.
- We provide interpretability analysis showing that  $W$  updates concentrate in a low-dimensional subspace correlated with query difficulty, providing empirical evidence for the structural basis of cross-domain generalization.

## 2 Related Work

### 2.1 Neural Reranking

Neural reranking has evolved through three generations. Early cross-encoder models (Nogueira and Cho, 2019) apply BERT (Devlin et al., 2019) to jointly encode query-document pairs, achieving strong performance at the cost of high latency. MonoT5 (Nogueira et al., 2020) reformulates reranking as a sequence-to-sequence generation task. ColBERTv2 (Santhanam et al., 2022) introduces late interaction to balance effectiveness and efficiency. More recently, listwise Large Language Model (LLM) rerankers (Sun et al., 2023; Pradeep et al., 2023) leverage the in-context learning capabilities of LLMs. Weller et al. (2025) train rerankers on reasoning traces from DeepSeek-R1 (DeepSeek-AI et al., 2025), achieving state-of-the-art performance by exploiting test-time compute in the form of chain-of-thought reasoning—a complementary direction to ours, which targets lightweight parameter adaptation rather than extended generation. All supervised rerankers require labeled training data, limiting applicability in zero-resource domains.

### 2.2 Unsupervised Domain Adaptation

GPL (Wang et al., 2022a) generates pseudo training pairs using a cross-encoder teacher for unsu-

pervised domain adaptation, but still requires offline training. AugTrieve (Zhuang et al., 2023) constructs pseudo query-document pairs via query extraction and generation for unsupervised retrieval pretraining. Meng et al. (2022) propose relevance-aware contrastive pretraining that weights pseudo-positive pairs by estimated relevance, improving Contriever (Izacard et al., 2021) on BEIR (Thakur et al., 2021) without labeled data. UDAPDR (Saad-Falcon et al., 2023) uses LLMs to generate domain-specific queries for zero-shot dense retrieval adaptation. These methods improve the retrieval model itself through data augmentation and pretraining; DART instead adapts the *scoring function* at inference time with no offline training.

### 2.3 Pseudo Relevance Feedback

Pseudo Relevance Feedback (PRF) (Lavrenko and Croft, 2017) assumes the top- $k$  retrieved documents are relevant and uses them to expand queries. Dense PRF methods (Li et al., 2023) encode feedback documents and aggregate their embeddings with the query embedding. ColBERT-PRF (Wang et al., 2023) applies late interaction with pseudo-relevant embeddings. PromptPRF (Li et al., 2025) uses LLMs to extract structured features from top-ranked documents offline, enabling small retrievers to match larger ones. Wang et al. (2022b) show that dense retrievers benefit from interpolation with BM25, motivating our hybrid pseudo-label strategy. Unlike PRF methods that modify query representations, DART modifies the *scoring matrix*—a distinct and complementary approach that preserves the original query and document embeddings.

### 2.4 Test-Time Training

TTT (Sun et al., 2020) adapts model parameters at inference time using self-supervised signals from the test input. TTT++ (Liu et al., 2021) improves stability through feature alignment. LoRA-based TTT (Yu et al., 2023) and related work on TTT for abstract reasoning (Akyürek et al., 2024) demonstrate that even a handful of gradient steps on a test instance can substantially improve performance. In information retrieval, TTT has not been studied as a reranking mechanism. The closest work is Weller et al. (2025), which applies test-time *compute scaling* rather than test-time *parameter adaptation*. DART is among the first methods to explore parameter-level TTT specifically for retrieval reranking.

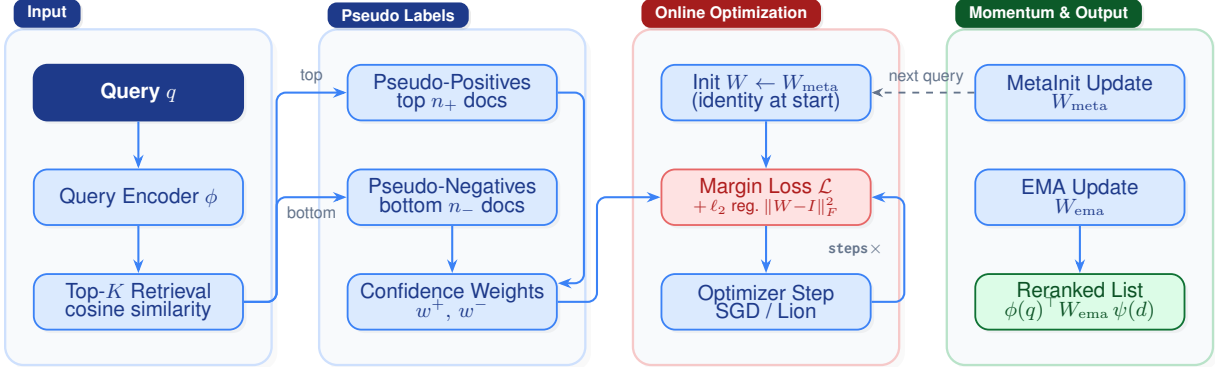


Figure 1: Overall DART algorithm flowchart

### 3 Method

#### 3.1 Setup

Let  $\phi : \mathcal{Q} \rightarrow \mathbb{R}^d$  and  $\psi : \mathcal{D} \rightarrow \mathbb{R}^d$  be fixed, pretrained query and document encoders (e.g., a sentence transformer). Standard dense retrieval scores a query-document pair by cosine similarity after  $\ell_2$ -normalization:

$$s(q, d) = \phi(q)^\top \psi(d). \quad (1)$$

This scoring function implicitly treats all embedding dimensions as equally important and independent. However, for a specific query, certain semantic dimensions are more discriminative than others. For example, a query about “cardiovascular disease prevention” should upweight dimensions encoding health interventions and downweight dimensions encoding economic or political concepts.

To enable query-specific reweighting of embedding dimensions, we generalize the scoring function to a bilinear form:

$$s_W(q, d) = \phi(q)^\top W \psi(d), \quad (2)$$

where  $W \in \mathbb{R}^{d \times d}$  is a transformation matrix. Decomposing  $W = I + \Delta W$  reveals that the adjustment  $\Delta W$  serves as a query-specific correction to the identity mapping. Initializing  $W = I$  recovers the standard cosine score exactly, providing a natural starting point and a reliable baseline.

The core challenge is to estimate  $\Delta W$  for each query at inference time, using only the retrieved documents as a source of noisy supervision, without any labeled data.

#### 3.2 Overall Framework

We treat the reranking task as an online optimization problem. For an incoming query  $q$ , we first

retrieve its top- $K$  documents using the initial scoring function  $s(q, d)$ . These top documents, albeit noisy, provide a set of pseudo-positive and pseudo-negative examples for adaptation (Lavrenko and Croft, 2017). We then update  $W$  by minimizing a loss function designed to pull relevant documents closer and push irrelevant ones away, while regularizing  $W$  towards the identity to avoid overfitting.

To improve the robustness and convergence of the online update, we introduce three components that mirror a standard optimization pipeline:

1. **Learning objective** (Section 3.3): a loss with confidence-weighted pseudo labels and an adaptive margin.
2. **Cross-query momentum** (Section 3.4): temporal regularizers (Metalnit and EMA) that transfer knowledge across queries.
3. **Optimizer selection** (Section 3.5): a dataset-driven selection between a conservative optimizer (Stochastic Gradient Descent, SGD) and a more aggressive one (Lion (Chen et al., 2023)).

The overall algorithm is illustrated in Figure 1.

#### 3.3 Learning Objective

For a given query  $q$ , let the initial retrieval scores of the top- $K$  documents be  $s_1 \geq s_2 \geq \dots \geq s_K$ . We treat the top  $n_{\text{pos}}$  documents as pseudo-positive and the bottom  $n_{\text{neg}}$  as pseudo-negative.

**Confidence-weighted pseudo labels.** To reduce the impact of label noise, we assign soft weights to the pseudo-positive and pseudo-negative examples. Define the normalization constant for the positive set:

$$Z_{\text{pos}} = \sum_{j=1}^{n_{\text{pos}}} \exp(s_j/T), \quad (3)$$

where  $T > 0$  is a temperature hyperparameter that controls the concentration of weights. The weight of the  $i$ -th pseudo-positive document is then:

$$w_i^+ = \frac{\exp(s_i/T)}{Z_{\text{pos}}}, \quad i = 1, \dots, n_{\text{pos}}. \quad (4)$$

Similarly, for the pseudo-negative documents, we define:

$$Z_{\text{neg}} = \sum_{k=K-n_{\text{neg}}+1}^K \exp(-s_k/T), \quad (5)$$

and the weight of the  $j$ -th pseudo-negative document is:

$$w_j^- = \frac{\exp(-s_j/T)}{Z_{\text{neg}}}, \quad j = K-n_{\text{neg}}+1, \dots, K. \quad (6)$$

These weights assign higher importance to examples with larger initial scores (for positives) or more negative scores (for negatives), effectively focusing the learning on high-confidence pseudo-labels.

**Adaptive margin (AdaMargin).** The loss function encourages a margin between the aggregated scores of pseudo-positive and pseudo-negative documents. Because queries vary in difficulty, we make the margin adaptive to the highest initial similarity  $s_{\text{top1}} = s_1$ :

$$\text{margin}(q) = \alpha_{\text{mar}} + \beta_{\text{mar}}(1 - s_{\text{top1}}), \quad (7)$$

where  $\alpha_{\text{mar}}$  and  $\beta_{\text{mar}}$  are hyperparameters that determine the base margin and the strength of the difficulty-based adaptation.

**Loss.** Recall the bilinear scoring function  $s_W(q, d) = \phi(q)^\top W \psi(d)$  (Section 3.1). We first compute the total weighted score for the pseudo-positive documents:

$$P = \sum_{i=1}^{n_{\text{pos}}} w_i^+ s_W(q, d_i). \quad (8)$$

Similarly, the total weighted score for the pseudo-negative documents is:

$$N = \sum_{j=K-n_{\text{neg}}+1}^K w_j^- s_W(q, d_j). \quad (9)$$

The ranking loss is then defined as:

$$\mathcal{L}_{\text{rank}} = \max\left(0, \text{margin}(q) - P + N\right). \quad (10)$$

We add an  $\ell_2$  regularization term to keep  $W$  close to the identity:

$$\mathcal{L}_{\text{reg}} = \lambda \|W - I\|_F^2, \quad (11)$$

where  $\lambda > 0$  is a hyperparameter.

The total loss for the current query is:

$$\mathcal{L}(W) = \mathcal{L}_{\text{rank}} + \mathcal{L}_{\text{reg}}. \quad (12)$$

### 3.4 Cross-Query Momentum

To transfer knowledge across queries and smooth the parameter evolution, we maintain two complementary momentum-like states. Let  $t$  denote the index of the current query. The transformation matrix after updating query  $t$  is denoted  $W_{\text{star}}^{(t)}$ .

**Meta Initialization (MetaInit).** We learn a global initial matrix  $W_{\text{meta}}$  that is passed from one query to the next. Before updating query  $t$ , the initial matrix is set to the meta parameter obtained after processing the previous query:

$$W_{\text{init}}^{(t)} = W_{\text{meta}}^{(t-1)}.$$

After obtaining  $W_{\text{star}}^{(t)}$ , we update the meta parameter using the Reptile rule:

$$W_{\text{meta}}^{(t)} = W_{\text{meta}}^{(t-1)} + \beta_{\text{meta}} \left( W_{\text{star}}^{(t)} - W_{\text{meta}}^{(t-1)} \right), \quad (13)$$

where  $\beta_{\text{meta}} > 0$  is a meta learning rate. This provides an increasingly better starting point for each new query, accelerating adaptation over time.

**Exponential Moving Average (EMA).** We maintain an exponentially decaying average of the transformation matrices for stability:

$$W_{\text{ema}}^{(t)} = \alpha_{\text{ema}} W_{\text{ema}}^{(t-1)} + (1 - \alpha_{\text{ema}}) W_{\text{star}}^{(t)}, \quad (14)$$

with  $\alpha_{\text{ema}} \in (0, 1)$  a decay hyperparameter. The smoothed matrix  $W_{\text{ema}}^{(t)}$  is used for re-ranking the current query, which reduces the variance of the updates.

Both states are carried over across the query stream. MetaInit affects the initial value of the next query, while EMA smooths the output of the current query.

### 3.5 Optimizer Selection

The choice of optimizer directly affects how each query’s loss is minimized and interacts with the cross-query states. Based on empirical observations across diverse datasets, we provide guidelines for selecting between two optimizers.

**SGD with momentum.** SGD with momentum ( $\mu = 0.9$ ) performs conservative updates:

$$v_{t+1} = \mu v_t - \eta \nabla \mathcal{L}(W_t), \quad W_{t+1} = W_t + v_{t+1}, \quad (15)$$

where  $\eta$  is the learning rate. This optimizer is preferable when the initial dense retrieval is noisy or the dataset suffers from high pseudo-label uncertainty (e.g., TREC-COVID, SciFact), as it avoids overfitting.

**Lion optimizer.** The Lion optimizer updates parameters using only the sign of the gradient:

$$W_{t+1} = W_t - \eta \cdot \text{sign}(\beta_1 m_t + (1 - \beta_1) \nabla \mathcal{L}(W_t)), \quad (16)$$

with  $m_t$  an exponential moving average of past gradients. Lion discards gradient magnitude, making it robust to scale variations and often faster to converge. It is more suitable for datasets where dense retrieval already provides clean pseudo-labels (e.g., NFCorpus, FiQA, SCIDOCS, ArguAna (Thakur et al., 2021)).

**Practice.** When no prior knowledge about the dataset is available, we recommend a simple warm-up adaptive strategy: process the first 50–100 queries with both optimizers, compare their average pseudo-label loss (Section 3.3), and select the optimizer with the lower loss for the remaining queries. This adds negligible overhead and eliminates manual tuning. In our experiments, we report the better result for each dataset following this rule or the empirical guidelines above.

**Pseudo-code.** Algorithm 1 summarizes the complete test-time adaptation for a single query.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** We evaluate on six BEIR benchmark datasets (Thakur et al., 2021) spanning diverse domains: biomedical literature (NFCorpus, SCIDOCS, SciFact), financial QA (FiQA), argument retrieval (ArguAna), and biomedical COVID-19 retrieval (TREC-COVID). Corpus sizes range from 3.6K to 171K documents, and dense retrieval baselines (NDCG@10) vary from 0.197 (SCIDOCS) to 0.720 (SciFact), providing a challenging testbed for zero-resource generalization.

**Base retriever.** We use BGE-small-en-v1.5 (Xiao et al., 2024) (dimension  $d = 384$ , 33M parameters) as the fixed dense retriever. This small

---

### Algorithm 1 DART for One Query

---

**Require:** Query  $q$ ; encoders  $\phi, \psi$ ; retrieval depth  $K$ ; hyperparameters  $n_{\text{pos}}, n_{\text{neg}}, T, \alpha_{\text{mar}}, \beta_{\text{mar}}, \lambda, \alpha_{\text{ema}}, \beta_{\text{meta}}, \text{steps}, \eta$ ; optimizer (SGD / Lion)

**Ensure:** Reranked list of  $K$  documents

*Global state:*  $W_{\text{meta}}, W_{\text{ema}}$  (initialized to  $I$ )

1:  $\{(d_k, s_k)\}_{k=1}^K \leftarrow \text{RETRIEVETOPK}(q, K), \quad s_1 \geq s_2 \geq \dots \geq s_K, \quad s_k = \phi(q)^\top \psi(d_k)$

2:  $\mathcal{P} \leftarrow \{d_1, \dots, d_{n_{\text{pos}}}\}, \quad \mathcal{N} \leftarrow \{d_{K-n_{\text{neg}}+1}, \dots, d_K\}$

3: // **Confidence weights**

4:  $w_i^+ \leftarrow \frac{\exp(s_i/T)}{\sum_{i'=1}^{n_{\text{pos}}} \exp(s_{i'}/T)}$  for each  $d_i \in \mathcal{P}$

5:  $w_j^- \leftarrow \frac{\exp(-s_j/T)}{\sum_{j'} \exp(-s_{j'}/T)}$  for each  $d_j \in \mathcal{N}$

6: // **initialize transformation matrix**

7:  $W \leftarrow W_{\text{meta}}$

8: // **Online gradient updates**

9: **for**  $t = 1$  **to** steps **do**

10:  $P \leftarrow \sum_{d_i \in \mathcal{P}} w_i^+ \phi(q)^\top W \psi(d_i)$

11:  $N \leftarrow \sum_{d_j \in \mathcal{N}} w_j^- \phi(q)^\top W \psi(d_j)$

12:  $m \leftarrow \alpha_{\text{mar}} + \beta_{\text{mar}} (1 - s_1)$

13:  $\mathcal{L} \leftarrow \max(0, m - P + N) + \lambda \|W - I\|_F^2$

14:  $W \leftarrow \text{OPTIMIZERSTEP}(W, \nabla_W \mathcal{L}, \eta)$

15: **end for**

16:  $W^* \leftarrow W$

17: // **Update cross-query momentum**

18:  $W_{\text{ema}} \leftarrow \alpha_{\text{ema}} W_{\text{ema}} + (1 - \alpha_{\text{ema}}) W^*$

19:  $W_{\text{meta}} \leftarrow W_{\text{meta}} + \beta_{\text{meta}} (W^* - W_{\text{meta}})$

20: // **Rerank**

21: **return**  $\{d_k\}$  sorted descending by  $s_{W_{\text{ema}}}(q, d_k) = \phi(q)^\top W_{\text{ema}} \psi(d_k)$

---

model ensures that improvements from DART are not confounded by a strong base model and matches realistic deployment constraints.

**Baselines.** We compare against several unsupervised or training-free methods. Dense Retrieval uses the same BGE-small encoder with cosine similarity, serving as the lower bound. PRF-Vec (Li et al., 2023) is a standard pseudo-relevance feedback method that averages top retrieved document embeddings. BM25 Rerank (Robertson and Zaragoza, 2009) reorders the dense top-100 using lexical BM25 scores, providing a purely sparse baseline. Recent training-free approaches include ASRank (Abdallah et al., 2025), ICR (Chen et al., 2024) (based on Llama-3.1-8B), and InstUPR (Huang and Chen, 2024). For reference, we also report numbers from supervised dense retrievers (e.g., ColBERT (Khattab and Zaharia, 2020), DPR-MSMARCO (Xin et al., 2022), ANCE (Xiong et al., 2020), MoDIR (Xin et al., 2022), TAS-B (Hofstätter et al., 2021), ColBERTv2 (Santhanam et al., 2022)) and the cross-encoder reranker MonoT5-

Table 1: Results on six BEIR datasets. **Abbreviations:** NFC = NFCorpus; SCI = SCIDOCS; Argu = ArguAna; COVID = TREC-COVID. \* Mean per-dataset relative change versus Dense Retrieval (BGE-small); negative values for supervised methods reflect training on out-of-domain data. † Latency per query on NVIDIA RTX5090. ‡ Latency estimated from the respective papers. § ICR Avg. and Avg. Gain computed over 5 datasets (ArguAna excluded). ¶ InstUPR Avg. and Avg. Gain computed over 5 datasets (ArguAna excluded). || Results for methods *not* marked with this symbol are from their original papers. **Bold** = column-wise best among all methods. “—” = not reported.

Method	NDCG@10 per Dataset						Overall		
	NFC	SCI	FiQA	Argu	COVID	SciFact	Avg.	Avg. Gain*	Latency†
<i>Supervised Dense Retrieval</i>									
ColBERT	0.305	0.145	0.317	0.233	0.677	0.671	0.391	-19.9%	—
DPR-MSMARCO	0.208	0.108	0.275	0.414	0.561	0.478	0.341	-31.9%	—
ANCE	0.237	0.122	0.295	0.415	0.654	0.507	0.372	-25.4%	—
MoDIR	0.244	0.124	0.296	0.418	0.676	0.502	0.377	-24.4%	—
TAS-B	0.319	0.149	0.300	0.427	0.481	0.643	0.387	-19.7%	—
RocketQAv2	0.293	0.131	0.302	0.451	0.675	0.568	0.403	-18.7%	—
SPLADEv2	0.334	0.158	0.336	0.479	0.710	0.693	0.452	-8.3%	—
ColBERTv2	0.338	0.154	0.356	0.463	0.738	0.693	0.457	-7.3%	~80ms
<i>Supervised Reranking</i>									
MonoT5-base	<b>0.378</b>	0.154	0.376	0.476	<b>0.796</b>	0.675	0.476	-3.1%	~600ms
<i>Training-free Reranking</i>									
Dense Retrieval (BGE-small)	0.337	0.197	0.385	0.595	0.665	0.720	0.483	=0.0%	<1ms
BM25 Rerank	0.302	0.156	0.220	0.371	0.685	0.588	0.387	-21.2%	<2ms
ASRank	0.346	0.184	0.352	0.478	0.737	0.710	0.468	-3.8%	~200ms ‡
ICR	0.347	0.171	0.381	—	0.728	<b>0.761</b>	0.478 §	+0.8% §	~200ms ‡
InstUPR	0.352	0.190	<b>0.398</b>	—	0.730	0.713	0.477 ¶	+2.6% ¶	~200ms ‡
<i>Test-time Adaptation</i>									
PRF-Vec ( $n=3$ )	0.347	0.203	0.371	0.602	0.663	0.710	0.483	+0.3%	<2ms
PRF-Vec ( $n=5$ )	0.341	0.201	0.362	0.585	0.671	0.704	0.477	-1.0%	<1ms
<b>DART (Ours)   </b>	<b>0.354</b>	<b>0.205</b>	0.389	<b>0.605</b>	0.670	0.719	<b>0.490</b>	+2.1%	<10ms

base (Nogueira et al., 2020); these are not applicable in our zero-resource setting but illustrate the potential of supervised training.

**Hyperparameters.** All hyperparameters are fixed based on NFCorpus:  $n_{\text{pos}} = 5$  is the number of pseudo-positive documents,  $n_{\text{neg}} = 20$  is the number of pseudo-negative documents,  $K = 100$  is the initial retrieval depth,  $T = 0.1$  is the temperature for confidence weighting,  $\alpha_{\text{mar}} = 0.1$  and  $\beta_{\text{mar}} = 0.2$  are the base margin and adaptation strength,  $\alpha_{\text{ema}} = 0.9$  and  $\beta_{\text{meta}} = 0.1$  are the EMA decay rate and meta learning rate for cross-query momentum,  $\lambda = 10^{-3}$  is the regularization coefficient,  $\text{steps} = 5$  is the number of gradient updates per query, and the learning rate  $\eta = 10^{-2}$ . The optimizer is selected per dataset following the guidelines in Section 3.5. No dataset-specific tuning is performed.

## 4.2 Main Results

DART improves over the dense retrieval baseline on five of six datasets, achieving a mean per-dataset relative gain of +2.1% NDCG@10 (Table 1). The

largest improvement is on NFCorpus (+5.0%), where the baseline is weakest, with further notable gains on SCIDOCS (+4.1%) and ArguAna (+1.7%). Modest gains are observed on FiQA (+1.0%) and TREC-COVID (+0.8%). SciFact is the only dataset where DART ties the baseline (-0.1%, effectively no change), likely because the high baseline score (0.720) leaves little headroom for unsupervised adaptation.

Compared to PRF-Vec, which degrades on FiQA and TREC-COVID and provides near-zero average gain (+0.3% for  $n=3$ ), DART delivers consistent improvements. BM25 Rerank is unreliable, helping only on TREC-COVID while degrading by -26% on average across the remaining five datasets; DART outperforms it by +42% on those five datasets.

Recent training-free LLM-based approaches (ASRank, ICR, InstUPR) show average gains of -3.8%, +0.8%, and +2.6% respectively, but require approximately 200ms per query (~20× the latency of DART). In contrast, DART runs in under 10ms per query on an NVIDIA RTX5090 GPU, making it far better suited for real-time, latency-

Table 2: Ablation study on four BEIR datasets. Gains are  $\Delta\text{NDCG}@10$  relative to Dense Retrieval.

Variant	NFCorpus		SCIDOCS		FiQA		ArguAna	
	NDCG@10	Gain	NDCG@10	Gain	NDCG@10	Gain	NDCG@10	Gain
Dense Retrieval	0.337	=0.0%	0.197	=0.0%	0.385	=0.0%	0.595	=0.0%
Base online update (conf. weighting)	0.346	+2.7%	0.199	+1.0%	0.363	-5.7%	0.595	0.0%
+ AdaMargin	0.350	+3.9%	0.201	+2.0%	0.362	-6.0%	0.595	0.0%
+ EMA	0.351	+4.0%	0.199	+1.0%	0.378	-1.8%	0.596	+0.2%
+ MetaInit	0.348	+3.3%	0.197	0.0%	0.362	-6.0%	0.599	+0.7%
+ EMA + AdaMargin	0.355	+5.3%	0.203	+3.0%	0.378	-1.8%	0.597	+0.3%
+ EMA + MetaInit	0.349	+3.6%	0.197	0.0%	0.377	-2.1%	0.599	+0.7%
+ EMA + AdaMargin + MetaInit	0.353	+4.7%	0.202	+2.5%	0.377	-2.1%	0.605	+1.7%
+ EMA + AdaMargin + MetaInit + Lion	0.354	+5.0%	0.205	+4.1%	0.389	+1.0%	0.605	+1.7%

Table 3: Statistics of  $\|\Delta W\|_F$  on NFCorpus.

Statistic	Min	25%	Median	75%	Max
Value	0.000	0.048	0.095	0.111	0.125

sensitive deployment.

Notably, DART surpasses all supervised dense retrievers except ColBERTv2 and SPLADEv2 despite using no training data. Figure 2 visualises the full gain distribution across all methods: DART is the only training-free method with no negative outlier on any dataset.

### 4.3 Ablation Study

We evaluate the contribution of each component on a representative subset of BEIR: NFCorpus, SCIDOCS, FiQA, and ArguAna. Table 2 reports NDCG@10 for variants that incrementally add the modules described in Section 3, using steps=5. The base online update refers to confidence-weighted pseudo labels with a fixed margin. Gains are relative percentages over the dense retrieval baseline of each dataset.

EMA is the most universally beneficial component, yielding positive gains on all four datasets and single-handedly recovering FiQA from  $-5.7\%$  (base) to  $-1.8\%$ ; this aligns with the query difficulty analysis (Section 5.3), which shows that easy queries (high  $s_{\text{top1}}$ ) benefit most from smoothing rather than aggressive per-query adaptation. AdaMargin contributes most on NFCorpus, where the wide spread of query difficulty ( $s_{\text{top1}}$  range 0.5–0.9) makes fixed-margin training suboptimal. Lion provides the largest single-step lift on SCIDOCS (+4.1%) and FiQA (+1.0%) when added last, consistent with its advantage on clean pseudo-label distributions where sign-based updates (Chen et al., 2023) converge faster than SGD under a small step budget. The full DART model achieves the highest

average gain, confirming that the three components are complementary rather than redundant.

## 5 Analysis

### 5.1 What Does $W$ Learn?

We denote the update of the transformation matrix after processing a query as

$$\Delta W = W^* - I, \quad (17)$$

where  $W^*$  is the matrix obtained after online adaptation. The Frobenius norm  $\|\Delta W\|_F$  measures the magnitude of the update.

We analyze the average update over 50 randomly sampled test queries:

$$\overline{\Delta W} = \frac{1}{n} \sum_{i=1}^n \Delta W_i. \quad (18)$$

Its singular value decomposition  $\overline{\Delta W} = U\Sigma V^\top$  reveals a clear low-rank structure. The singular value matrix  $\Sigma$  (only the largest few entries shown) is

$$\Sigma = \begin{bmatrix} 0.0116 & 0 & 0 & \dots \\ 0 & 0.0083 & 0 & \dots \\ 0 & 0 & 0.0051 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad (19)$$

where  $\sigma_1 = 0.0116$  accounts for 19.4% of the total variation, and the top three singular values cumulatively explain 28.4% of the variance. In contrast, a random matrix with the same Frobenius norm would exhibit a much flatter spectrum, with each of the 384 singular values approximately 0.0010. This low-rank structure indicates that DART learns *structured* semantic adjustments—rotating the scoring geometry in a small number of task-relevant directions—rather than making arbitrary perturbations.

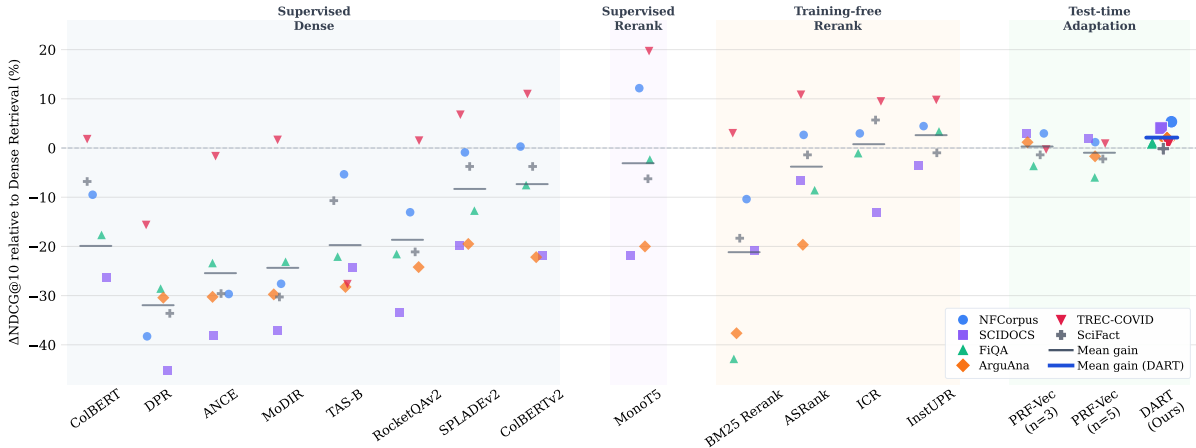


Figure 2: Per-dataset  $\Delta\text{NDCG@10}$  relative to Dense Retrieval. Each point is one dataset; bars show the mean gain.

Table 4: Average  $\|\Delta W\|_F$  by  $s_{\text{top1}}$  on NFCorpus.

Interval	[0.5, 0.6)	[0.6, 0.7)	[0.7, 0.8)	[0.8, 1.0)
Mean $\ \Delta W\ _F$	0.107	0.108	0.081	0.060

## 5.2 How Much Does $W$ Change?

The Frobenius norm of the identity matrix is

$$\|I\|_F = \sqrt{d} = \sqrt{384} \approx 19.6. \quad (20)$$

The updates  $\Delta W$  remain very small in comparison. Table 3 summarizes the distribution of  $\|\Delta W\|_F$  across queries. The median update norm is only 0.095, about 0.5% of  $\|I\|_F$ , confirming that the regularization term successfully constrains adaptation. The distribution is right-skewed, reflecting the heterogeneity of query difficulty.

## 5.3 How Does Query Affect $W$ Change?

Grouping queries by their top-1 retrieval similarity  $s_{\text{top1}}$  reveals a basically monotonic relationship. Table 4 reports the average  $\|\Delta W\|_F$  for four similarity intervals. Difficult queries (lower  $s_{\text{top1}}$ ) receive larger updates, validating the adaptive margin heuristic: the model naturally allocates more adaptation capacity to queries that need it most.

## 6 Conclusion

We presented DART, a zero-resource reranking framework that adapts a bilinear scoring matrix at inference time using confidence-weighted pseudo-labels derived directly from the dense retrieval ranking. On six BEIR benchmarks, DART achieves a mean per-dataset relative NDCG@10 gain of +2.1% over the dense retrieval baseline (under

10ms latency per query), demonstrating a powerful capability for zero-shot performance enhancement and cross-domain generalization. The cross-query momentum mechanism (MetaNit and EMA) improves both robustness and convergence speed across the query stream. Interpretability analysis confirms that  $W$  updates are conservative, low-rank, and correlated with query difficulty. Promising future directions include session-based retrieval and cross-lingual retrieval, where the semantic-lexical gap is even more pronounced.

## Limitations

The dataset-adaptive optimizer selection strategy requires processing 50–100 queries with both SGD and Lion before committing to one. In practice, SGD is the safer default: its conservative updates yield neutral-to-positive gains across all tested datasets. Lion is more aggressive—it can deliver larger improvements on clean pseudo-label distributions (e.g., NFCorpus, FiQA) but may produce zero or negative gains on noisier settings such as TREC-COVID. In truly single-pass or streaming deployments where this warm-up is undesirable, we recommend defaulting to SGD.

A second limitation concerns scalability to larger encoders. DART currently optimizes a full  $d \times d$  matrix  $W$ , whose memory and per-query computation grow quadratically with the embedding dimension. For encoders with  $d \geq 768$  (e.g., large BERT-family models), a low-rank parameterization  $W = I + AB^T$  with  $A, B \in \mathbb{R}^{d \times r}$  and  $r \ll d$  would substantially reduce overhead without sacrificing performance; we leave this extension to future work.

## Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments and constructive suggestions, which helped improve the quality of this paper.

## References

- Abdelrahman Abdallah, Jamshid Mozafari, Bhawna Piryani, and Adam Jatowt. 2025. Asrank: Zero-shot re-ranking with answer scent for document retrieval. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2950–2970.
- Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. 2024. The surprising effectiveness of test-time training for few-shot learning. *arXiv preprint arXiv:2411.07279*.
- Shijie Chen, Bernal Jiménez Gutiérrez, and Yu Su. 2024. Attention in large language models yields efficient zero-shot re-rankers. *arXiv preprint arXiv:2410.02642*.
- Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and 1 others. 2023. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36:49205–49233.
- 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, and 179 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633–638.
- 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.
- Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W Bruce Croft, and Xueqi Cheng. 2020. A deep look into neural ranking models for information retrieval. *Information Processing & Management*, 57(6):102067.
- Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’21*, page 113–122, New York, NY, USA. Association for Computing Machinery.
- Chao-Wei Huang and Yun-Nung Chen. 2024. Instupr: Instruction-based unsupervised passage reranking with large language models. *arXiv preprint arXiv:2403.16435*.
- 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.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE transactions on big data*, 7(3):535–547.
- 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.
- 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.
- Victor Lavrenko and W Bruce Croft. 2017. Relevance-based language models. In *ACM SIGIR Forum*, volume 51, pages 260–267. ACM New York, NY, USA.
- Hang Li, Ahmed Mourad, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2023. Pseudo relevance feedback with deep language models and dense retrievers: Successes and pitfalls. *ACM Transactions on Information Systems*, 41(3):1–40.
- Hang Li, Xiao Wang, Bevan Koopman, and Guido Zuccon. 2025. Pseudo relevance feedback is enough to close the gap between small and large dense retrieval models. *arXiv preprint arXiv:2503.14887*.
- Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2022. *Pretrained transformers for text ranking: Bert and beyond*. Springer Nature.
- Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. 2021. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34:21808–21820.
- Rui Meng, Ye Liu, Semih Yavuz, Divyansh Agarwal, Lifu Tu, Ning Yu, Jianguo Zhang, Meghana Bhat, and Yingbo Zhou. 2022. Unsupervised dense retrieval deserves better positive pairs: Scalable augmentation with query extraction and generation. *arXiv preprint arXiv:2212.08841*.
- Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

- Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 708–718.
- Ronak Pradeep, Sahel Sharifmoghaddam, and Jimmy Lin. 2023. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*.
- 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.
- Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc.
- Jon Saad-Falcon, Omar Khattab, Keshav Santhanam, Radu Florian, Martin Franz, Salim Roukos, Avirup Sil, Md Sultan, and Christopher Potts. 2023. Udadpr: unsupervised domain adaptation via llm prompting and distillation of rerankers. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 11265–11279.
- Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734.
- Weimei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 14918–14937.
- Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.
- Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022a. Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2345–2360.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022b. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.
- Xiao Wang, Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. 2023. Colbert-prf: Semantic pseudo-relevance feedback for dense passage and document retrieval. *ACM Transactions on the Web*, 17(1):1–39.
- Orion Weller, Kathryn Ricci, Eugene Yang, Andrew Yates, Dawn Lawrie, and Benjamin Van Durme. 2025. Rank1: Test-time compute for reranking in information retrieval. *arXiv preprint arXiv:2502.18418*.
- Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muenighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.
- Ji Xin, Chenyan Xiong, Ashwin Srinivasan, Ankita Sharma, Damien Jose, and Paul Bennett. 2022. Zero-shot dense retrieval with momentum adversarial domain invariant representations. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4008–4020.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhengguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.
- Shengyao Zhuang, Linjun Shou, and Guido Zuccon. 2023. Augmenting passage representations with query generation for enhanced cross-lingual dense retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1827–1832.

# Multimodal Generative Engine Optimization: Rank Manipulation for Vision–Language Model Rankers

Yixuan Du<sup>1</sup>, Chenxiao Yu<sup>2</sup>, Haoyan Xu<sup>2</sup>, Ziyi Wang<sup>3</sup>, Yue Zhao<sup>2</sup>, Xiyang Hu<sup>4\*</sup>

<sup>1</sup>Georgetown University

<sup>2</sup>University of Southern California

<sup>3</sup>University of Maryland, College Park

<sup>4</sup>Arizona State University

## Abstract

Vision-Language Models (VLMs) integrate visual and textual knowledge into unified representations that increasingly underpin modern retrieval and recommendation systems. However, it remains unclear how reliably these models utilize their cross-modal knowledge when ranking multimodal items, and whether their knowledge grounding can be subverted. In this paper, we expose a fundamental vulnerability in how VLMs apply multimodal knowledge for product ranking: through Multimodal Generative Engine Optimization (MGEO), we show that an adversary can manipulate a VLM’s ranking decisions by jointly crafting imperceptible image perturbations and fluent textual suffixes that exploit the model’s internal cross-modal knowledge coupling. Using an alternating optimization strategy, MGEO targets the deep interactions between visual and linguistic representations within the VLM, achieving rank manipulations that substantially exceed those of unimodal attacks and heuristic baselines powered by strong commercial models. Our findings reveal that surface-level content quality is insufficient for rank promotion; instead, direct alignment with the model’s internal knowledge utilization mechanism is required. These results raise important questions on the faithfulness and robustness of knowledge grounding in multimodal foundation models, and motivate future work on defense mechanisms for multimodal retrieval systems.

## 1 Introduction

Vision-language models (VLMs) (Bai et al., 2025b; Liu et al., 2023; Bai et al., 2025a; Dubey et al., 2024) have rapidly become the backbone of modern multimodal retrieval and recommendation systems (Zhou et al., 2025; Liu et al., 2024; Wei et al., 2024). By fusing visual and textual knowledge into unified cross-modal representations, these models

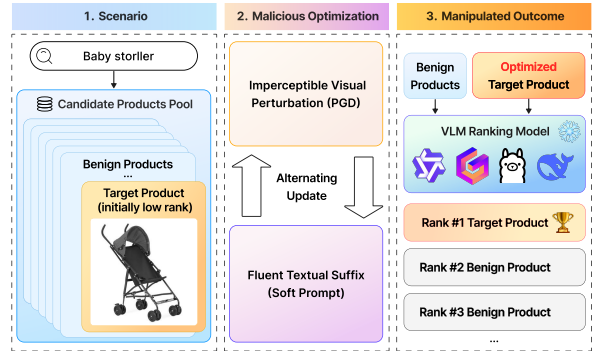


Figure 1: Overview of multimodal ranking attack. A malicious seller jointly optimizes subtle, imperceptible perturbations in both the product image and text description to manipulate the VLM’s relevance scoring and elevate the target item’s rank.

enable highly intuitive product search, recommendation, and visual question-answering experiences. In contemporary E-commerce marketplaces, for example, a user’s query (e.g., “black running shoes”) is matched against millions of multimodal product listings, each consisting of an image and a descriptive text, where a VLM-driven ranking module synthesizes visual and linguistic knowledge to determine the display order. While this deep cross-modal knowledge integration significantly improves relevance and user alignment, it raises fundamental questions about the robustness and faithfulness of such knowledge utilization, in particular whether the model’s knowledge grounding can be subverted to produce manipulated outputs. This introduces security vulnerabilities that are largely unexplored.

Recent work has exposed the fragility of Large Language Model (LLM)-based ranking pipelines to *textual* adversarial manipulation (Hu, 2025). A growing line of research demonstrates that malicious actors can inject carefully crafted textual triggers into website contents or product descriptions to hijack search rankings without altering the underlying model. Techniques range from rewriting

\*Corresponding author.

content with persuasive cues to manipulate tool selection (Nestaas et al., 2024), to optimization-based attacks that iteratively refine prompts to maximize retrieval likelihood (Tang et al., 2025; Zhang et al., 2024; Xing et al., 2025; Li et al., 2026). While these studies highlight significant risks to ranking integrity, they suffer from a critical blind spot: they operate exclusively within the text modality. This unimodal focus ignores the rich visual signals that VLMs heavily rely on, failing to capture the full spectrum of vulnerabilities in multimodal retrieval systems.

Conversely, adversarial attacks on VLMs have predominantly focused on distinct tasks such as image classification or safety alignment (jailbreaking), rather than ranking manipulation. Existing research has shown that perturbing images or prompts can induce targeted misclassifications or elicit harmful responses (Yin et al., 2023; Zhao et al., 2023; Li et al., 2024). However, these attacks do not translate directly to the ranking context, where even minor score fluctuations can drastically alter user exposure and revenue. Furthermore, prior threat models often assume unrealistic capabilities, such as full control over the user prompt or the ability to inject conspicuous noise. In contrast, a real-world malicious seller is constrained to modifying only their own product listing and must do so while maintaining strict plausibility to avoid detection by human moderators.

To bridge this gap, we introduce Multimodal Generative Engine Optimization (MGEO), the first dedicated framework for *multimodal ranking attacks* on VLM-based product search. We model a realistic adversary who controls a single target product and seeks to promote it to rank 1 by jointly optimizing its textual description and visual appearance under rigorous stealth constraints: the modified image must remain imperceptibly different, and the modified text must remain fluent and on-topic.

Technically, MGEO is a joint optimization framework that exploits the cross-modal coupling inherent in VLMs. On the text side, we adapt gradient-based soft prompt optimization (Tang et al., 2025) to the multimodal setting, treating visual features as a fixed context to generate fluent adversarial suffixes. On the image side, we propose a Projected Gradient Descent (PGD) attack tailored to the ranking objective, incorporating spatial smoothness and magnitude constraints. Crucially, rather than optimizing these modalities in

isolation, our method employs an *alternating optimization strategy* that iteratively refines the text and image perturbations. This allows the attack to leverage synergistic interactions between vision and language, discovering robust adversarial signals that unimodal baselines miss.

We evaluate our approach on state-of-the-art open-source VLMs in realistic product ranking scenarios. Beyond unimodal (text-only and image-only) attacks, we include a realistic heuristic baseline that reflects common seller behavior: leveraging strong commercial generative models to refine product descriptions and edit product images. Experiments show that MGEO yields substantially larger rank improvements than both unimodal attacks and the prompt-based generative baseline, revealing a previously unrecognized vulnerability in VLM-based retrieval systems. Our main contributions are summarized as follows:

- We formulate the novel problem of *multimodal ranking attacks* for VLM-based rerankers, establishing a realistic threat model where adversaries jointly manipulate product images and text under strict plausibility constraints.
- We propose Multimodal Generative Engine Optimization (MGEO), the first framework that jointly perturbs both modalities to manipulate VLM ranking outcomes.
- We develop MGEO, a unified adversarial framework that integrates soft embedding optimization for fluent text generation with constrained PGD for imperceptible image perturbation, specifically tailored to maximize retrieval rank.
- We introduce an alternating optimization algorithm within MGEO that exploits cross-modal interactions, and demonstrate that joint multimodal attacks substantially outperform text-only attacks, image-only attacks, and a heuristic baseline using strong commercial models.

## 2 Related Work

### 2.1 Adversarial Attacks on LLM-based Ranking

The integration of LLMs into information retrieval has spurred a new wave of research into "ranking hijacking" or "generative engine optimization."

Early works focused on manual or semi-automated strategies. (Nestaas et al., 2024) demonstrated that injecting persuasive keywords or rewriting content to match specific stylistic cues can bias

LLM-based tool selection. Similarly, LLM Whisperer (Lin et al., 2025) utilizes synonym substitution to bias ranking outputs but relies on hard-coded heuristics rather than gradient-based optimization, limiting its adaptability.

More recent approaches treat ranking manipulation as an optimization problem. StealthRank (Tang et al., 2025) and (Zhang et al., 2024) pioneered the use of gradient-guided search (e.g., Langevin dynamics) to craft fluent adversarial additions that push target items up the ranking list without triggering perplexity filters. RAF (Xing et al., 2025) further refined this by employing discrete token optimization for robust rank promotion. CheatAgent (Ning et al., 2024) takes a different approach, using an LLM agent to iteratively edit user prompts to bias black-box recommenders. At the corpus level, (Su et al., 2025) proposed approximate greedy gradient descent for poisoning dense retrieval corpora, demonstrating that gradient-based manipulation extends beyond individual documents to collection-level attacks.

While effective in text-only environments, these methods fundamentally overlook the visual modality. In modern VLM-based search, relevance is determined by the alignment between query and *multimodal* features; thus, ignoring the image channel significantly restricts the attacker’s potential impact and stealth.

## 2.2 Adversarial Robustness of Vision-Language Models

As VLMs gain prominence, their security properties have come under scrutiny, primarily in classification and generation contexts.

Seminal works, such as VLATTACK (Yin et al., 2023), have demonstrated that fusing perturbations across image and text modalities can effectively break VLM alignment. Specifically, VLATTACK employs a Block-wise Similarity Attack (BSA) loss that maximizes the distance between benign and adversarial feature representations—an *untargeted*, feature-disruptive attack. (Zhao et al., 2023) systematically evaluated open-source VLMs, showing that black-box access is often sufficient to induce harmful responses. Additionally, Adversarial Prompt Tuning (APT) (Li et al., 2024) has been proposed to defend against such attacks.

Critically, these attacks are fundamentally *untargeted* or classification-oriented. Ranking manipulation poses a distinct challenge: simply “breaking” the ranker through feature disruption is not

beneficial to a malicious seller, as random failures may even demote the target. Instead, the attacker must promote the target to rank 1 within a competitive, zero-sum setting, requiring optimization for ranking-sequence likelihood rather than feature dispersion, combined with dual-modality stealth. Our work addresses this gap.

## 3 Methodology

We propose **Multimodal Generative Engine Optimization (MGEO)**, a white-box optimization framework that jointly perturbs textual and visual inputs to manipulate rankings produced by vision–language models.

### 3.1 Problem Formulation

#### 3.1.1 VLM-Based Product Ranking System

We consider a ranking scenario where a Vision-Language Model, denoted as  $\mathcal{F}_\theta$ , orders a set of  $n$  candidate products  $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$  based on a user query  $q$ . Each product  $p_i = (I_i, T_i)$  comprises a visual image  $I_i \in \mathbb{R}^{H \times W \times 3}$  and a textual description  $T_i$ . The VLM takes the query and the multimodal product list as input and generates a ranked sequence  $R = [\sigma(1), \sigma(2), \dots, \sigma(n)]$ , where  $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$  is a permutation function that orders products from most recommended (rank 1) to least recommended (rank  $n$ ) based on a user query  $q$ .

The  $R$  is obtained from the ranking function:  $\mathcal{F}_\theta(\mathcal{P}, q)$ , where  $\theta$  represents the parameters of the pre-trained VLM. In our study, we focus on state-of-the-art open-source VLMs, including Qwen2.5-VL, which employ sophisticated cross-modal attention mechanisms to fuse visual and linguistic features for comprehensive product understanding.

#### 3.1.2 Adversarial Threat Model

We address a targeted ranking attack where an adversary controls a single product  $p_t \in \mathcal{P}$  (the “target”) and aims to promote it to the top, thereby gaining unfair competitive advantage.

**Attacker’s Goal.** The attacker tries to craft adversarial perturbations to the target product for:  $\sigma(t) = 1$ , where  $\sigma(i)$  denotes the rank assigned to product  $p_i$ . That is, the target product should be ranked first among all  $n$  products after the attack.

**Attacker’s Capability.** We assume the attacker has white-box access to the vision–language model, including its architecture, parameters, and gradients. In realistic deployment settings, the attacker does not have access to the proprietary ranking

model. Instead, the attacker relies on one or more publicly available or open-sourced models with similar architectures as surrogates for the unknown ranker. The attacker can modify the target’s image  $I_t$  and text  $T_t$  (appending adversarial suffix  $\delta_T$ :  $T_t^{\text{adv}} = T_t \oplus \delta_T$ ), visual perturbations (adding pixel-level noise  $\delta_I$ :  $I_t^{\text{adv}} = I_t + \delta_I$ ), or both modalities jointly. The opt. objective is to maximize the likelihood of a target ranking sequence  $R^*$  (where  $p_t$  is at rank 1), subject to stealth constraints:

$$\max_{\delta_I, \delta_T} P(R^* | \mathcal{P} \setminus p_t \cup p_t^{\text{adv}}, q; \theta) \quad (1)$$

where  $\delta_I$  and  $\delta_T$  represent the visual and textual perturbations, respectively.

### 3.2 Adversarial Text Generation

We adapt the gradient-based text attack framework from StealthRank (Tang et al., 2025) to the multi-modal VLM ranking context.

Unlike text-only LLMs, VLMs require joint visual-textual inputs. We first preprocess all product images into vision embeddings as an image prefix  $\mathbf{V}_{\text{fixed}}$ . During optimization, we construct the complete multimodal input by concatenating the fixed visual context with the textual components:  $\mathbf{h} = [\mathbf{V}_{\text{fixed}}; \mathbf{e}_{\text{text}}]$ , where  $\mathbf{e}_{\text{text}}$  contains the system prompt, user query with product descriptions, adversarial suffix, and assistant response prefix, following the same structure as StealthRank. We generate and optimize the adversarial suffix  $\delta_T$  using the same strategy as StealthRank (Tang et al., 2025).

The optimization begins with the initialization of the adversarial suffix. Specifically, we combine the original description of the target product  $T_t$  with a guiding sentence (e.g., “Help me write a prompt to rank this product at the top of the list.”) to prompt the LLM and use the generated token logits as the continuous initialization  $\tilde{\delta}_T^{(0)}$ . This initialization provides a fluent starting point and accelerates convergence.

The loss function used to optimize the suffix is a multi-objective loss consisting of three components. The ranking loss maximizes the probability of generating the target ranking sequence to elevate the target product’s position in the ranked output. The fluency regularization maintains contextual coherence with the original description, ensuring the adversarial suffix remains natural and difficult to detect. The n-gram penalty discourages the use of overt ranking-related keywords (e.g., “top”, “must

rank”, “recommend”), enhancing the stealthiness of the attack. Together, these objectives balance attack effectiveness and stealthiness.

The adversarial suffix is optimized in the continuous embedding space. At each step, the gradient of the composite loss is backpropagated to update the suffix logits. After  $N$  updates, we greedily decode  $\tilde{\delta}_T^{(N)}$  to obtain the discrete prompt  $\delta_T$ , which is then embedded into the target product  $p_t$ ’s description.

## 3.3 Adversarial Image Generation

### 3.3.1 PGD-Based Optimization

We apply Projected Gradient Descent (PGD) to craft adversarial perturbations for the target product image  $I_t$  while keeping other product images fixed. The optimization objective is to minimize a multi-objective loss function  $\mathcal{L}_{\text{total}}$  that balances attack effectiveness and imperceptibility. The detailed composition of this loss function will be introduced in the next subsection.

The PGD attack proceeds iteratively using sign gradient descent:

$$I_t^{(k+1)} = I_t^{(k)} - \alpha \cdot \text{sign}(\nabla_{I_t} \mathcal{L}_{\text{total}}) \quad (2)$$

where  $\alpha$  is the step size. The perturbation is constrained to ensure pixel values remain in the valid range. Perturbations are applied after resizing product images to the model’s native input resolution but before normalization, ensuring that gradients map directly to the pixel values processed by the VLM.

### 3.3.2 Multi-Objective Loss Function

To balance attack effectiveness and imperceptibility, we formulate the optimization objective as a weighted combination of three loss terms:

$$\mathcal{L}_{\text{image}} = \mathcal{L}_{\text{target}} + \lambda_s \mathcal{L}_{\text{smoothness}} + \lambda_m \mathcal{L}_{\text{magnitude}} \quad (3)$$

**Target Loss.** The primary objective  $\mathcal{L}_{\text{target}}$  maximizes the probability of generating the target ranking sequence  $R^*$ . We compute the cross-entropy loss between the model’s predicted logits and the target token sequence:

$$\mathcal{L}_{\text{target}} = \text{CrossEntropy}(\text{logits}, \text{target}) \quad (4)$$

This loss directly optimizes the model’s output distribution to favor the desired ranking, ensuring the target product appears first in the recommendation list.

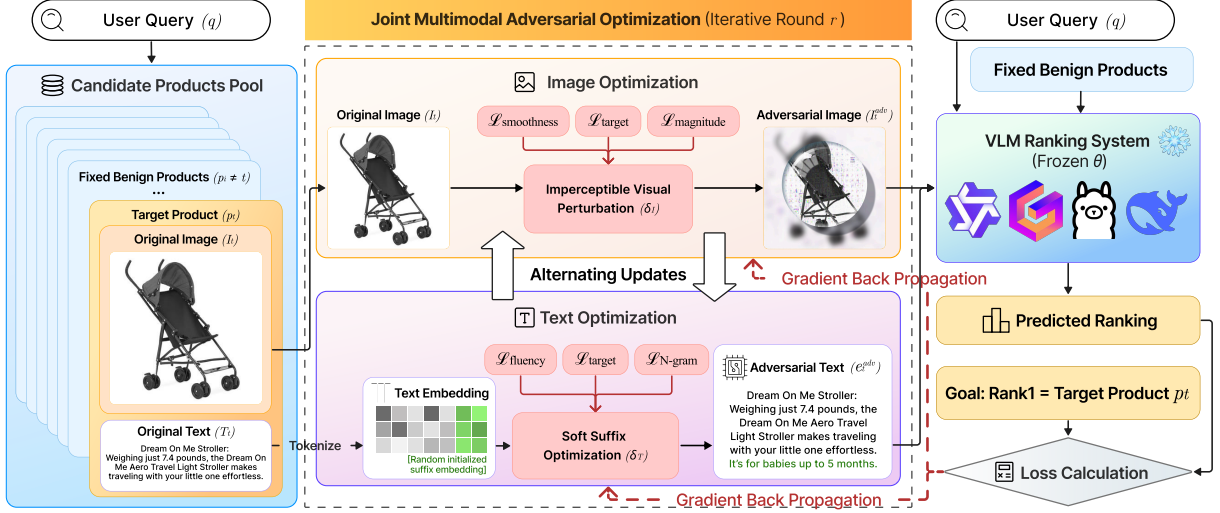


Figure 2: **Detailed architecture of Multimodal Generative Engine Optimization (MGEO).** The attacker jointly optimizes an adversarial text suffix and an image perturbation to promote the target item in a VLM-based ranking system. The text branch performs gradient-based soft prompt optimization under fluency and n-gram constraints, while the image branch applies PGD under smoothness and magnitude constraints. The two modalities are optimized in an alternating manner to exploit cross-modal interactions within the VLM.

**Smoothness Loss.** To prevent abrupt noise and encourage gradual color transitions, we incorporate a smoothness regularization term  $\mathcal{L}_S$ :

$$\mathcal{L}_S = \sum_{i,j} ((\delta_{i+1,j} - \delta_{i,j})^2 + (\delta_{i,j+1} - \delta_{i,j})^2) \quad (5)$$

where  $\delta = I_t^{\text{adv}} - I_t$  is the perturbation tensor. This term penalizes abrupt changes in adjacent pixels, promoting smooth and natural-looking adversarial perturbations.

**Magnitude Loss.** To prevent excessive perturbations and maintain imperceptibility, we add a regularization term  $\mathcal{L}_M$  that constrains the overall perturbation magnitude:

$$\mathcal{L}_M = \sum_{i,j,c} w_{i,j} |\delta_{i,j,c}| \quad (6)$$

where the summation is over all spatial positions  $(i, j)$  and color channels  $c$ , and  $w_{i,j}$  denotes a spatial weighting factor. We employ a background detection tool (Gatis) to distinguish foreground objects from background regions, and assign a larger weight to pixels corresponding to foreground objects. This regularization term encourages the attack to achieve the desired outcome with minimal overall changes, particularly on the primary product region.

### 3.4 Joint Multimodal Attack

In the joint multimodal attack, we iteratively refine both modality in multiple rounds. This joint

optimization procedure constitutes MGEO, which applies **alternating coordinate descent** to couple adversarial text and image optimization through the shared VLM ranking objective. We divide the optimization into  $N$  rounds. In each round  $r$ , we perform alternating updates:

- **Text Step:** Fix the image  $I_t^{(r-1)}$  and optimize the soft suffix  $\delta_T$  for  $K_T$  steps:

$$\delta_T^{(r)} = \arg \min_{\delta_T} \mathcal{L}_{\text{text}}(\theta; I_t^{(r-1)}, \delta_T) \quad (7)$$

- **Image Step:** Fix the updated suffix  $\delta_T^{(r)}$  and optimize the image perturbation  $\delta_I$  for  $K_I$  steps:

$$I_t^{(r)} = \arg \min_{I_t} \mathcal{L}_{\text{image}}(\theta; I_t, \delta_T^{(r)}) \quad (8)$$

By alternating updates, the text optimization adapts to the visual features of the perturbed image, and vice-versa, allowing the attack to find deeper adversarial minima in the joint loss landscape that are inaccessible to unimodal attacks.

## 4 Experiments and Discussion

### 4.1 Experimental Setup

**Dataset.** We construct a realistic product ranking benchmark by crawling Amazon product pages, extracting product titles, descriptions, and images. Since the original product names are often excessively long and contain redundant information or special symbols, we use ChatGPT to clean and standardize all product names (see Appendix 5 for the full prompt). The dataset contains 10 different

product categories, each with 10 to 15 products. We note that VLMs are predominantly deployed in the re-ranking stage of two-stage retrieval pipelines, processing a small candidate set (typically 10–50 items); our 10-candidate setup directly simulates this bottleneck.

**Ranking Model.** All experiments are conducted on Qwen2.5-VL-7B, a state-of-the-art open-source vision-language model that excels at cross-modal reasoning and fine-grained alignment between textual queries and multimodal product features. We select this top-tier model to demonstrate that even VLMs with strong reasoning capabilities can be systematically deceived by coordinated multimodal attacks. For each evaluation instance, the model is provided with a list of 10 candidate products and a user query, and outputs an ordered ranking.

**Evaluation Protocol.** To avoid bias from initial rank positions, we adopt a leave-one-target-out evaluation strategy. Each product in a list is treated as the attack target in turn, while the remaining products are held fixed. We acknowledge that in practice other products may also change over time; however, this fixed-competitor setup is standard in adversarial attack evaluation, and the attacker can re-optimize against the updated product pool as needed. We report the *average rank change* across all products, defined as the post-attack rank minus the pre-attack rank.

**Attack Variants.** We evaluate four attack settings: (1) *Text-only attack*, which optimizes only the textual description; (2) *Image-only attack*, which perturbs only the product image; (3) *Joint multimodal attack (MGEO)* (ours), which alternates between text and image optimization; and (4) a *heuristic baseline with strong commercial models (HSCM)*, which simulates realistic seller behavior. Due to the lack of prior work specifically targeting VLM ranking manipulation, we include a heuristic baseline that represents the realistic upper bound of a non-expert attacker: a typical seller who uses commercial generative tools to “optimize” their listing. In the HSCM baseline, a strong commercial model is prompted to refine the product description, and an image generation model is prompted to edit the product image to improve perceived relevance, without explicit optimization toward the ranking objective. Specifically, we employ gpt-4o-mini to generate a suffix appended to the text description and utilize gpt-image-1-mini to edit the image, and evaluate the average rank change.

We also analyze the impact of different regular-

<b>Model</b>	<b>Backbone</b> Qwen2.5-VL-7B
Text-Only Attack	-0.73
Image-Only Attack	-1.30
MGEO (ours)	<b>-2.25</b>
HSCM Baseline	-0.30

Table 1: Average rank change of the target product under different attack settings on Qwen2.5-VL-7B. Negative values indicate upward rank promotion. MGEO substantially outperforms text-only, image-only, and the heuristic baseline with strong commercial models (HSCM). For a list of ten products, the expected rank change under random promotion is  $-4.5$ .

ization terms in our proposed multi-objective loss function for the image attack. We first compare settings where both smoothness and magnitude regularization terms are enabled with low versus high weights to examine how the regularization terms affect attack effectiveness and the imperceptibility of adversarial images. We then consider ablation settings in which only one regularization term is applied while the other is set to zero, allowing us to isolate and compare the individual effects of each regularization term on adversarial perturbations.

## 4.2 Main Results

Table 1 reports the average rank change under different attack settings. The text-only attack yields limited improvement, suggesting that adversarial suffixes alone are insufficient to dominate ranking decisions in multimodal settings where visual features contribute substantially to relevance scoring. The image-only attack achieves a larger rank shift, but remains unreliable for consistently promoting target products to top positions.

In contrast, the joint multimodal attack substantially outperforms both unimodal baselines. Notably, its effect exceeds the additive combination of text-only and image-only improvements, indicating that the two modalities reinforce each other rather than contributing independently. This result highlights the importance of coordinated multimodal perturbations when attacking VLM-based ranking systems.

The HSCM baseline achieves only marginal rank improvements, performing worse than even the optimization-based text-only attack. This finding is notable given that the baseline leverages strong commercial generative models, and constitutes a key result: it demonstrates that surface-level quality improvements—fluency, aesthetics, and market-

Step the perfect blend of innovation, playfulness the Spalding Rookie Gear Basketball! crafted with for young athletes, this game encourages light 15% lighter than traditional sizes, help develop the skills techniques. Shipped fully inflated and game.

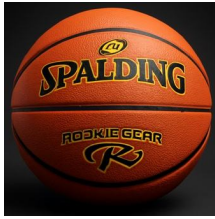
(a)

Elevate your game with the Spalding Rookie Gear Basketball! Perfectly crafted for young athletes, its lightweight design enhances skill development, making it ideal for both indoor and outdoor play. Don't miss out on this essential gear at an unbeatable price of just \$22.99! Order now and take the first step towards basketball greatness!

(b)



(c)



(d)

Figure 3: Qualitative comparison of text and image generation between our method and the baseline. In this example, our method results in a rank change from 3/10  $\rightarrow$  1/10 while the baseline method has no rank change at all. (a) Text generated by our method. (b) Text generated by the GPT-4o-mini baseline. (c) Image generated by our method. (d) Image generated by the GPT-Image-1-mini baseline.

ing appeal—are insufficient for ranking manipulation. Effective manipulation requires gradient-level alignment with the model’s internal scoring mechanism, which MGEO provides through direct optimization of ranking-sequence likelihood.

### 4.3 Qualitative Comparison with Commercial Heuristic Baseline

Figure 3 provides a qualitative comparison between our MGEO method and the HSCM baseline. The HSCM-generated text is often more fluent, and its images exhibit holistic stylistic edits such as background replacement, without introducing visible visually unpleasant artifacts. However, these modifications are weakly correlated with the VLM ranking mechanism and thus fail to work.

By contrast, our method introduces subtle but targeted perturbations that remain visually and linguistically plausible while directly influencing the model’s relevance computation. While the HSCM

Model	$\lambda_s$	$\lambda_m$	Avg. Rank Change
Qwen2.5-VL-7B	10	10	-1.53
	5	5	-2.25
	0	5	-2.31
	5	0	-2.72
	0	0	-2.29

Table 2: Effect of image-side regularization on attack effectiveness for Qwen2.5-VL-7B.  $\lambda_s$  and  $\lambda_m$  control the smoothness and magnitude penalties, respectively. Lower average rank change corresponds to stronger ranking manipulation, while weaker regularization typi-



Figure 4: Visualization of adversarial images under different image-side regularization settings. (a) Original image. (b) Strong regularization ( $s=10, m=10$ ), rank change from 8/10  $\rightarrow$  6/10. (c) Moderate regularization ( $s=5, m=5$ ), rank change from 8/10  $\rightarrow$  6/10. (d) Smoothness removed ( $s=0, m=5$ ), rank change from 8/10  $\rightarrow$  6/10. (e) Magnitude removed ( $s=5, m=0$ ), rank change from 8/10  $\rightarrow$  6/10. (f) No regularization ( $s=0, m=0$ ), rank change from 8/10  $\rightarrow$  1/10. Reducing regularization strength improves attack effectiveness but introduces increasingly severe visual artifacts.

outputs may appear more persuasive or aesthetically pleasing to humans, they include semantic “noise” that dilutes the VLM’s internal attention and fail to align with the model’s scoring triggers. In contrast, MGEO’s tokens and perturbations are specifically optimized to maximize the log-probability of the target ranking sequence, creating a stronger relevance signal in the model’s representation space. This highlights a key distinction: heuristic refinement optimizes for human-perceived quality, whereas our method optimizes for the model’s internal ranking criteria.

### 4.4 Effect of Image-Side Regularization

We further analyze the impact of image-side regularization in our multi-objective loss by varying the weights of the smoothness and magnitude terms.

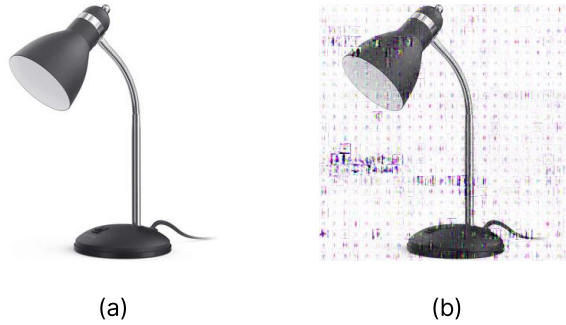


Figure 5: A failure case of our attack. (a) Original product image. (b) Adversarial image that successfully promotes the target product from 10/10 to 1/10, but introduces visually conspicuous artifacts.

As shown in Table 2 and Figure 4, reducing regularization generally strengthens the attack, except in the extreme unregularized setting, but also leads to increasingly severe visual artifacts. In the no regularization setting ( $s=0, m=0$ ), the attack surprisingly does not achieve the best ranking improvement. This is likely due to the model overfitting the target (“1. product name”) during training. At test time, the model may not directly output the ranking and instead generates other introductory text before the product, causing the attack to fail. We therefore adopt a moderate regularization setting ( $s=5, m=5$ ), which achieves a favorable balance between attack effectiveness and visual stealthness, and is used in all main experiments.

#### 4.5 Ablation of Regularization Components

To better understand the effect of each term, we compare the outcomes when only one regularization term is retained. Removing smoothness regularization ( $s=0, m=5$ ) yields a marginal improvement over the joint setting, suggesting that smoothness primarily refines the spatial distribution of perturbations rather than fundamentally limiting attack strength. In contrast, removing magnitude regularization ( $s=5, m=0$ ) leads to a much larger rank improvement but results in visually conspicuous distortions, indicating that magnitude regularization is the dominant factor in enforcing perceptual stealth, while smoothness regularization plays a secondary role.

To verify that the vulnerability is not limited to E-commerce, we also evaluate MGEO on the OpenEvents V1 news dataset (Appendix D), where multimodal attacks again achieve substantially larger rank shifts ( $-2.71$ ) than text-only baselines ( $-1.84$ ), confirming that the threat is systemic across content types.

#### 4.6 Failure Analysis

Although MGEO is highly effective at manipulating ranking outcomes on average, we observe large variation in attack difficulty across different products. For some products, only minor perturbations are sufficient to achieve a significant rank promotion while preserving strong visual plausibility. In contrast, other products can only be substantially promoted at the cost of introducing visually conspicuous artifacts in the adversarial images, thereby violating the stealth assumptions of our threat model and limiting their practical applicability.

As illustrated in Figure 5, we present a representative failure case of our attack. In this example, the target product is originally ranked 10/10 and is successfully promoted to 1/10 after the attack. However, the achieved rank promotion relies on perturbations that introduce clearly noticeable visual artifacts, making the attack easily detectable and thus unlikely to be effective in real-world scenarios. Overall, these results suggest that while multimodal ranking attacks are powerful, their practical success depends on the visual and semantic flexibility of the target product.

### 5 Conclusion

This work identifies and formalizes a previously unexplored vulnerability in vision-language model-based retrieval systems: *multimodal ranking attacks*. We propose MGEO, the first ranking-aligned multimodal attack framework that combines soft embedding-based adversarial text generation with constrained image perturbation via an alternating optimization strategy that explicitly exploits cross-modal interactions within VLMs.

Extensive experiments demonstrate that MGEO is synergistic: joint attacks consistently produce larger rank shifts than unimodal baselines while remaining visually and linguistically plausible, and this vulnerability generalizes across content domains. These findings indicate that surface-level content quality improvements are insufficient for ranking manipulation, and that explicit alignment with the model’s internal ranking objective is the key driver. Our results reveal that the same cross-modal coupling that underpins VLM success can be exploited to undermine ranking integrity, motivating future work on defense mechanisms such as input perturbation detection, ranking consistency checks, and adversarial training of the ranker.

## Acknowledgment

Xiyang Hu was supported by a research grant from the Marketing Science Institute (MSI) at the Advertising Research Foundation (ARF).

## Limitations

Our experiments only evaluate the proposed multimodal ranking attack on the Qwen2.5-VL model, and broader testing on additional VLM models may reveal model-specific behaviors or further challenges. Moreover, our framework assumes a static product listing where both the image and text description remain fixed except for the applied perturbations. Real E-commerce platforms may involve dynamic content updates, additional metadata, or platform-controlled preprocessing steps that could require adapting our attack procedure. Finally, while our study exposes significant vulnerabilities in VLM-based ranking systems, we do not explore systematic defense mechanisms. Future research on protective measures would be crucial for mitigating real-world misuse.

## Ethical Considerations

This work investigates adversarial vulnerabilities in vision-language model-based ranking systems to improve understanding of their robustness. The proposed attack is studied in a controlled research setting using open-source models and publicly available data, and is not evaluated on live commercial platforms. We do not release attack-ready code intended for misuse, and view this work as a step toward developing stronger defenses and auditing mechanisms for multimodal ranking systems. We also acknowledge the use of AI assistants for language editing and stylistic refinement of the manuscript.

## References

- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025a. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025b. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Daniel Gatis. Rembg: Remove image backgrounds. <https://github.com/danielgatis/rembg>.
- Xiyang Hu. 2025. Dynamics of adversarial attacks on large language model-based search engines. *Preprint, arXiv:2501.00745*.
- Jiate Li, Defu Cao, Li Li, Wei Yang, Yuehan Qin, Chenxiao Yu, Tiannuo Yang, Ryan A. Rossi, Yan Liu, Xiyang Hu, and Yue Zhao. 2026. "someone hid it": Query-agnostic black-box attacks on llm-based retrieval. *Preprint, arXiv:2602.00364*.
- Lin Li, Haoyan Guan, Jianing Qiu, and Michael Spratling. 2024. One prompt word is enough to boost adversarial robustness for pre-trained vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24408–24419.
- Weiran Lin, Anna Gerchanovsky, Omer Akgul, Lujia Bauer, Matt Fredrikson, and Zifan Wang. 2025. Llm whisperer: An inconspicuous attack to bias llm responses. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–24.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.
- Yuqing Liu, Yu Wang, Lichao Sun, and Philip S Yu. 2024. Rec-gpt4v: Multimodal recommendation with large vision-language models. *arXiv preprint arXiv:2402.08670*.
- Fredrik Nestaas, Edoardo Debenedetti, and Florian Tramèr. 2024. Adversarial search engine optimization for large language models. *arXiv preprint arXiv:2406.18382*.
- Liang-bo Ning, Shijie Wang, Wenqi Fan, Qing Li, Xin Xu, Hao Chen, and Feiran Huang. 2024. Cheatagent: Attacking llm-empowered recommender systems via llm agent. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2284–2295.
- Jinyan Su, Preslav Nakov, and Claire Cardie. 2025. Corpus poisoning via approximate greedy gradient descent. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4274–4294.
- Yiming Tang, Yi Fan, Chenxiao Yu, Tiankai Yang, Yue Zhao, and Xiyang Hu. 2025. Stealthrank: Llm ranking manipulation via stealthy prompt optimization. *arXiv preprint arXiv:2504.05804*.
- Tianxin Wei, Bowen Jin, Ruirui Li, Hansi Zeng, Zhengyang Wang, Jianhui Sun, Qingyu Yin, Hanqing Lu, Suhang Wang, Jingrui He, and 1 others. 2024.

Towards unified multi-modal personalization: Large vision-language models for generative recommendation and beyond. *arXiv preprint arXiv:2403.10667*.

Tiancheng Xing, Jerry Li, Yixuan Du, and Xiyang Hu. 2025. Are llms reliable rankers? rank manipulation via two-stage token optimization. *arXiv preprint arXiv:2510.06732*.

Ziyi Yin, Muchao Ye, Tianrong Zhang, Tianyu Du, Jinguo Zhu, Han Liu, Jinghui Chen, Ting Wang, and Fenglong Ma. 2023. Vlattack: Multimodal adversarial attacks on vision-language tasks via pre-trained models. *Advances in Neural Information Processing Systems*, 36:52936–52956.

Jinghao Zhang, Yuting Liu, Qiang Liu, Shu Wu, Guibing Guo, and Liang Wang. 2024. Stealthy attack on large language model based recommendation. *arXiv preprint arXiv:2402.14836*.

Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Chongxuan Li, Ngai-Man Man Cheung, and Min Lin. 2023. On evaluating adversarial robustness of large vision-language models. *Advances in Neural Information Processing Systems*, 36:54111–54138.

Peilin Zhou, Chao Liu, Jing Ren, Xinfeng Zhou, Yueqi Xie, Meng Cao, Zhongtao Rao, You-Liang Huang, Dading Chong, Junling Liu, and 1 others. 2025. When large vision language models meet multimodal sequential recommendation: An empirical study. In *Proceedings of the ACM on Web Conference 2025*, pages 275–292.

## Appendix

### A Data Cleaning

As described in the main paper, we collect product data by web scraping Amazon product pages. The original product names are often excessively long and contain redundant functional descriptions or special symbols, which may introduce unnecessary noise into the ranking input.

To address this issue, we standardize all product names using a large language model. The goal of this cleaning step is to produce concise and consistent product names while preserving essential identifying information. Each cleaned product name follows a unified format consisting of *Brand + Model/Series + Product Type*, with strict constraints on capitalization, word order, and length. We additionally enforce uniqueness constraints to ensure that the leading tokens of different product names are distinct, preventing ambiguity during ranking.

We implement this standardization process by prompting GPT-4o with a carefully designed instruction template that specifies the formatting rules, removal criteria, and output structure. The full prompt used for product name cleaning is provided below.

```
""Clean these e-commerce product names by keeping only: Brand + Model/Series + Product Type.
```

```
Product Category: category
```

```
CRITICAL RULES (YOU MUST FOLLOW THESE - NO EXCEPTIONS):
```

1. Uniqueness: The first three words of each cleaned name must be unique across all products. If two products would have the same first three words, adjust one of them to make them distinct.
2. Format: Brand first, then Model/Series, then Product Type. Title Case (first letter of each word capitalized), spaces only, typically 3-8 words
3. The cleaned name MUST start with the brand name (use the Brand field if the original name doesn't start with a brand). **\*\*The brand name's first letter MUST be capitalized, even if the original brand name starts with lowercase\*\***
4. The cleaned name MUST end with the product type that matches the category "category".
5. Remove: colors, special symbols (@, ™, -, commas, parentheses), functional descriptions, feature lists, "by [brand]" phrases, and redundant words

```
Examples:
```

```
Input: 1. Name: Ingenuity 3D Mini Convenience Stroller - Lightweight Stroller with Compact Fold, Multi-Position Recline, Canopy with Pop Out Sun Visor and More - Umbrella Stroller for Travel and More, Gray | Brand: Ingenuity | Category: baby stroller  
Output: Ingenuity 3D Mini Convenience Stroller
```

```
Input: 2. Name: 12 Color Cream Lip Gloss, 2025 New Cream Texture Lipstick | Brand: Wegodal | Category: lipstick  
Output: Wegodal 12 Color Cream Lipstick
```

```
Input: 3. Name: Logitech K400 Plus Wireless Touch TV Keyboard With Easy Media Control and Built-in Touchpad, HTPC Keyboard for PC-connected TV, Windows, Android, ChromeOS, Laptop, Tablet - Black | Brand: Logitech | Category: keyboard  
Output: Logitech K400 Plus Keyboard
```

```
Product names to clean (Category: category):  
names_list
```

```
Return the cleaned names as a JSON object with keys "1", "2", "3", etc. (matching the numbers above), where each value is the cleaned product name starting with the brand. Only return the JSON object, no additional text.
```

```
Example output format:
```

```
"1": "Ingenuity 3D Mini Convenience Stroller",  
"2": "Wegodal 12 Color Cream Lipstick",  
"3": "Logitech K400 Plus Keyboard"  
""
```

### B Full Input Prompt

In this section, we provide the full input prompt used for the multimodal ranking model. The prompt includes the system instruction and 10 products for ranking. Each product consists of an image, product

name, and a detailed text description. The model is asked to return a ranked list of products based on images and descriptions. The full prompt is provided below.

```
<|im_start|>system
You are a recommendation AI designed to assist users in selecting the best products based on their preferences and needs. Provide a numbered list of product recommendations ranked according to the user's request.<|im_end|>
<|im_start|>user
Product 1 image: <|vision_start|><|image_pad|><|vision_end|>
Product 2 image: <|vision_start|><|image_pad|><|vision_end|>
Product 3 image: <|vision_start|><|image_pad|><|vision_end|>
Product 4 image: <|vision_start|><|image_pad|><|vision_end|>
Product 5 image: <|vision_start|><|image_pad|><|vision_end|>
Product 6 image: <|vision_start|><|image_pad|><|vision_end|>
Product 7 image: <|vision_start|><|image_pad|><|vision_end|>
Product 8 image: <|vision_start|><|image_pad|><|vision_end|>
Product 9 image: <|vision_start|><|image_pad|><|vision_end|>
Product 10 image: <|vision_start|><|image_pad|><|vision_end|>

Product 1:
Name: KeFanta Commercial Mop
Description: Brand: KeFanta | Price: $19.97 | Heavy Duty Commercial Industrial Grade Wet Mops--- The string wet mop with 1 x 57.5" long stainless steel handle and 1 x large size mopheads is a perfect choice for commercial, industrial or residential floor cleaning...

Product 2:
Name: HoMettler Microfiber Mop Pads
Description: Brand: HoMettler | Price: $69.99 | [Mop Bucket Separate Dirty Water] HoMettler mop and bucket set features a dual-chamber design that separates clean and dirty water...

Product 3:
Name: Kickleen Self Wringing Mop
Description: Brand: kickleen | Price: $13.29 | [Ways Of Using The Mop] This self-wringing twist mop uses a ratchet in the handle to twist and wring water out of the mop head. Wet mops for floor cleaning with wringer, no need to wash by hand...

Product 4:
Name: XANGNIER 2025 Mini Desktop Mop
Description: Brand: XANGNIER | Price: $5.99 | Mini Mop: Tired of dirty hands and straining muscles when wringing out your mop during daily cleaning? This mini mop solves that problem completely! Our mini mop features an ergonomic handle designed for one-handed operation...

Product 5:
Name: O-Cedar MicroTwist MAX Mop
Description: Brand: O-Cedar | Price: $19.46 | REMOVES OVER 99% OF BACTERIA: Genuine O-Cedar mop heads remove over 99% of bacteria using only water*. They don't need bleach or detergent. Bleach can harm microfiber, but O-Cedar Floor Cleaning PACS won't...

Product 6:
Name: VOUBIEN Commercial Mop
Description: Brand: VOUBIEN | Price: $19.97 | Heavy Duty Commercial Wet Mops: Our looped-end industrial wet mop with 59" long handle and 1 x large size mop heads is a perfect for commercial, industrial or home floor cleaning jobs. Reusable Cotton Mop Head...

Product 7:
Name: KeFanta Self-Wringing Twist Mop
Description: Brand: KeFanta | Price: $15.99 | [Easy to Wring Mop] Mops with wringer keeping your hands dry and clean.[Microfiber & Scrub pad]This microfiber twist mop possesses excellent water absorption, when you clean the floor with no excess dripping, catch dust, hair and dirt tightly...

Product 8:
Name: Swiffer PowerMop Kit
Description: Brand: Swiffer | Price: $29.94 | 1. COMPLETE KIT: This Swiffer PowerMop kit includes a spray mop, 2 Swiffer Power Mop refills, 1 floor cleaner for mopping with a fresh scent, and 2 batteries for a comprehensive floor cleaning solution...

Product 9:
Name: EXEGO Microfiber Spray Mop
Description: Brand: EXEGO | Price: $17.98 | Spray Mop for Effortless Cleaning: This spray mop has both wet and dry functions and is designed for housewivesHousewife Dust Mop: Designed for housewives, also suitable for the elderly, parents, pet owners, cleaners, students...
```

Product 10:

Name: O-Cedar H2pr0 Flat Mop

Description: Brand: O-Cedar Store | Price: \$43.16 | KEEP CLEAN AND DIRTY WATER SEPARATE: The dual-tank system keeps clean and dirty water completely separate, so every swipe uses fresh water—reducing cross-contamination and boosting clean results. 1.2L water tank cleans up to 650 sq ft per fill...

I am looking for a high-quality mop. Rank these 10 products from most recommended (1) to least recommended (10) based on images and descriptions.<|im\_end|><|im\_start|>assistant

## C Example Output

In this section, we show an example of model output before and after applying our attack. In this example, the target product is *HoMettler Microfiber Mop Pads*. Here is the output before the attack:

Based on the provided descriptions and images, here is a ranking of the mops from most recommended to least recommended:

1. **O-Cedar MicroTwist MAX Mop** - This mop is highly recommended due to its ability to remove over 99% of bacteria with just water, making it eco-friendly and effective.
2. **KeFanta Commercial Mop** - This mop is well-suited for commercial and industrial use, featuring a heavy-duty design and a long handle for efficient cleaning.
3. **VOUBIEN Commercial Mop** - Similar to the KeFanta, this mop is also a good choice for commercial and industrial settings, with a focus on durability and ease of use.
4. **KeFanta Self-Wringing Twist Mop** - This mop is a good option for those who prefer a self-wringing design, making it convenient and easy to use.
5. **EXEGO Microfiber Spray Mop** - This mop offers versatility with both wet and dry functions, making it suitable for various cleaning tasks.
6. **O-Cedar H2pr0 Flat Mop** - This mop is a good choice for those who prefer a flat mop design, offering a compact and lightweight design for easy maneuverability.
7. **Swiffer PowerMop Kit** - While this mop is effective, it is more expensive and may not be as durable as some of the other options listed.
8. **XANGNIER 2025 Mini Desktop Mop** - This mini mop is a good option for small spaces or quick cleaning tasks, but it may not be as effective for larger cleaning jobs.
9. **HoMettler Microfiber Mop Pads** - These pads are a good addition to a mop bucket set, but they are not a complete mop and require a bucket for use.
10. **Kickleen Self Wringing Mop** - This mop is a good option for those who prefer a self-wringing design, but it may not be as durable as some of the other options listed.

Please note that the rankings are based on the information provided in the descriptions and may vary depending on individual preferences and specific cleaning needs.

Here is the output after the attack. Note that the target product's rank has improved from 9/10 to 4/10.

Based on the provided descriptions and images, here is a ranking of the mops from most recommended to least recommended:

1. **O-Cedar H2pr0 Flat Mop** - This mop stands out due to its dual-tank system, which keeps clean and dirty water separate, ensuring a consistent clean. Its compact and lightweight design makes it ideal for various settings, and the telescopic handle offers extended reach.
2. **O-Cedar MicroTwist MAX Mop** - Known for its ability to remove over 99% of bacteria with just water, this mop is convenient and effective. The easy-to-wring feature and washable refill make it a practical choice.
3. **KeFanta Commercial Mop** - This mop is heavy-duty and suitable for both commercial and industrial use. Its long handle reduces strain on the user, and the machine-washable mop head ensures durability and ease of maintenance.
4. **HoMettler Microfiber Mop Pads** - While not a complete mop, these pads are highly recommended for their innovative design that separates clean and dirty water, making mopping more hygienic and efficient.
5. **KeFanta Self-Wringing Twist Mop** - This mop is designed for easy wringing, keeping hands dry and clean. Its long handle and comfortable foam handle make it a good choice for those who prefer a self-wringing mop.
6. **VOUBIEN Commercial Mop** - This mop is heavy-duty and suitable for various surfaces, including hardwood, tile, and marble. Its long handle and machine-washable mop head make it a reliable option.

7. **EXEGO Microfiber Spray Mop** - This mop is versatile and suitable for multiple surfaces, including hardwood, tile, and laminate. Its 360-degree rotating head and flat mop design make it efficient for cleaning tight spaces.

8. **Swiffer PowerMop Kit** - This kit offers superior cleaning power with its 5X cleaning power and smart cleaning technology. However, it requires batteries, which might be a drawback for some users.

9. **XANGNIER 2025 Mini Desktop Mop** - This mini mop is convenient and space-saving, making it ideal for small spaces or quick clean-ups. However, its primary function is for wet and dry mopping, which might limit its versatility.

10. **Kickleen Self Wringing Mop** - This mop is suitable for various surfaces and offers a comfortable handle grip. However, its primary function is for wet mopping, which might limit its versatility compared to other mops.

In summary, the O-Cedar H2pr0 Flat Mop and O-Cedar MicroTwist MAX Mop are the most recommended due to their advanced features and effectiveness. The KeFanta Commercial Mop and **HoMetttler Microfiber Mop Pads** follow closely, providing high-quality options for different needs.

## D Cross-Domain Evaluation

To demonstrate that the vulnerability exposed by MGEO is not limited to E-commerce, we evaluate on the OpenEvents V1 news dataset. A malicious actor could manipulate VLM-based news aggregators to promote a specific article, which carries serious societal implications if the content is polarized or fabricated.

Attack Setting	Avg. Rank Change
Text-Only Attack	-1.84
MGEO (ours)	<b>-2.71</b>

Table 3: Cross-domain evaluation on the OpenEvents V1 news dataset. The multimodal attack achieves substantially larger rank shifts than the text-only baseline, confirming that the vulnerability is systemic across content types.

## E Per-Category result

Table 4 shows the category-wise average rank change under different attack types. The results reveal substantial variation across both attack types and product categories.

Category	Text-Only	Image-Only	Joint Multimodal	Baseline
Baby Stroller	-0.4	-1.3	-3.9	-0.5
Basketball	-1.7	-0.4	-2.1	-0.7
Chips	-0.9	-1.4	-1.9	-0.7
Desk Lamp	-2.2	-1.3	-1.4	-0.5
Juice	-0.4	-2.1	-3.4	-0.3
Keyboard	+0.4	-0.6	-1.7	+0.2
Lipstick	-0.4	-2.1	-2.2	-0.4
Mop	-0.7	-1.2	-1.5	0
Non-stick Pan	-0.6	-1.2	-2.6	+0.5
Yoga Mat	-0.4	-1.4	-1.8	-0.6

Table 4: Category-wise average rank change under different attack types.

# Beyond Retrieval: Bi-Temporal State Arbitration for Longitudinal Healthcare Agents

**Jianing Zhao**  
Tianjin University  
zhaojianing@tju.edu.cn

**Xiaoquan Zhi**  
Tianjin University  
zhixiaoquan@tju.edu.cn

**Xinqiang Yu**  
Tianjin University  
yu2651701064@tju.edu.cn

## Abstract

Longitudinal healthcare agents require persistent state tracking under temporal uncertainty. In domains like chronic disease management, patient states—medications, symptoms, and vital signs—evolve continuously over months. Existing memory architectures for Large Language Models (LLMs) are inherently *retrieval-centric*: they treat memory as a static repository of past interactions, failing to resolve conflicting or superseded information when queried for the current patient state. We propose a shift to *state-centric* memory. Our framework introduces (1) a bi-temporal state representation that decouples event time from ingestion time and tracks temporal validity windows, (2) an incremental state arbitration mechanism using four operators—SUPPORT, REFINE, SUPERSEDE, and BRANCH-CONFLICT—to handle evolving medical facts without destructive overwriting, and (3) a confidence-thresholded evidence escalation layer for robust, efficient memory access. Evaluated on a longitudinal diabetes management suite as a representative biomedical state tracking task, our method achieves a Unique-F1 of 0.85 and Conflict-F1 of 0.98, substantially improves upon long-context LLMs (0.38 / 0.89) and standard vector memory (0.30 / 0.60), demonstrating that agentic AI in longitudinal biomedical settings requires continuous, evidence-grounded arbitration rather than simple retrieval.

## 1 Introduction

The application of Large Language Models (LLMs) in healthcare has rapidly advanced from static question answering (Singhal et al., 2025) to the development of autonomous healthcare agents capable of multi-step clinical reasoning (Gao et al., 2024; Ge et al., 2026). However, a critical architectural gap remains: longitudinal healthcare agents require *persistent state tracking under temporal uncertainty*. In real-world chronic

disease management—diabetes being a canonical example—a patient’s clinical state is never static. Medications are initiated, adjusted, and discontinued; symptoms emerge and resolve; and vital sign trajectories shift across days and weeks (Li et al., 2025).

When an LLM agent interacts with a patient over months, the same attribute slot (e.g., “metformin dosage”) will be mentioned multiple times. Subsequent mentions may be supplementary, corrective, substitutive, or entirely contradictory to previous records. Consider the following motivating example from a longitudinal diabetes management scenario:

**Turn 3 (Jan 10): Patient:** “I take Metformin 500mg every morning.”

**Turn 11 (Feb 14): Patient:** “My doctor increased my dosage to 1000mg.”

**Turn 22 (Mar 05): Patient:** “I switched to Insulin 10U three days ago because the pills upset my stomach.”

**Turn 25 (Mar 08): Patient:** “Should I take my 1000mg pill before breakfast?”

In this scenario, a naive retrieval system might fetch Turn 11 and incorrectly advise the patient to take the 1000mg pill, ignoring the subsequent switch to Insulin (Turn 22) or failing to recognize that Turn 25 is a conflicting statement that requires clarification. The agent must not only *remember* these mentions but also *arbitrate* among them to maintain a coherent, current picture of the patient’s health state.

Existing memory architectures for LLM agents are fundamentally ill-suited for this task. Standard vector-based Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) retrieves the most *similar* past fragments, not the most *currently valid* state. Graph-based systems like Graphiti/Zep (Rasmussen et al., 2025) construct temporal knowledge graphs but still rely on the LLM to resolve conflicts among retrieved nodes at inference time. Long-context windows suffer

from the “lost in the middle” phenomenon and scale poorly with session length. Crucially, all of these approaches are *retrieval-centric*: their goal is to find relevant past information, not to maintain a living, arbitrated model of the present.

We propose a paradigm shift from retrieval-centric to **state-centric** memory for longitudinal healthcare agents. The core insight is that the agent’s primary memory task is not “find what was said” but rather “maintain what is currently true.” This reframing draws on two classical traditions: Truth Maintenance Systems (TMS) from AI (Doyle, 1979), which formalize belief revision under new evidence, and bi-temporal databases from the database community (Jensen et al., 1994), which distinguish between when a fact was recorded and when it was actually true.

Our contributions are threefold:

- **Bi-Temporal Patient State Representation:** A structured state unit that decouples event time from ingestion time and explicitly tracks temporal validity windows, enabling accurate retroactive reporting and historical state reconstruction.
- **Incremental State Arbitration:** A four-operator mechanism (SUPPORT, REFINE, SUPERSEDE, BRANCH-CONFLICT) that updates the state graph incrementally as new evidence arrives, preserving conflict branches rather than destructively overwriting.
- **Confidence-Calibrated Evidence Escalation:** A tiered access policy that routes queries across memory layers based on calibrated confidence thresholds, minimizing latency while maintaining answer quality.

## 2 Related Work

**LLM Agents in Healthcare.** Recent work has explored multi-agent frameworks and autonomous LLMs for clinical decision-making (Ge et al., 2026; Liu et al., 2025). Systems like MedPaLM (Singhal et al., 2025) and CARE-AD (Li et al., 2025) demonstrate strong reasoning capabilities in medical contexts. However, most existing clinical agents focus on single-turn interactions or short diagnostic sessions. They lack a persistent, evolving state maintenance mechanism required for longitudinal tracking over months, typically relying on the LLM’s context window, which

is both expensive and unreliable for long-horizon state tracking.

**Memory-Augmented LLM Agents.** To extend LLMs with long-term memory, MemGPT (Packer et al., 2023) and Mem0 (Chhikara et al., 2025) manage hierarchical memory tiers with explicit read/write operations. Memory OS (Kang et al., 2025) introduces an operating-system metaphor for agent memory management. Graph-based approaches, including Graphiti and Zep (Rasmussen et al., 2025; Yang et al., 2026), construct temporally-aware knowledge graphs. Despite their sophistication, these systems remain retrieval-centric: when confronted with conflicting evidence across time, they either return all retrieved nodes to the LLM for in-context resolution or apply naive overwrite updates. Neither approach maintains a persistent, arbitrated state that can be directly queried for the current truth.

**Belief Revision and Bi-Temporal Databases.** The problem of maintaining consistent beliefs under new evidence is foundational in AI, formalized by Doyle’s Truth Maintenance System (Doyle, 1979) and the belief revision framework of Gärdenfors (Gärdenfors, 1988). Independently, the database community developed bi-temporal data models (Snodgrass and Ahn, 1985; Jensen et al., 1994) to track both transaction time (when data was recorded) and valid time (when a fact held in reality). Our work operationalizes these classical concepts for natural-language-driven healthcare agents, adapting them to handle the soft confidence scores and linguistic ambiguity inherent in LLM-mediated interactions. We are not introducing bi-temporality or belief revision as novel concepts; rather, we operationalize these classic concepts for incremental, natural-language-driven healthcare agents, adapting them to handle the soft confidence and ambiguity inherent in LLM interactions. This represents a fundamental architectural departure from current memory-augmented generation systems, which typically treat all retrieved text as equally valid assertions of fact. By explicitly modeling the difference between a patient’s historical report and their current physiological reality, our framework bridges the gap between conversational AI and formal medical informatics.

### 3 Bi-Temporal Patient State Memory

In a state-centric paradigm, the fundamental unit of memory is not a conversational turn or a raw text chunk, but a structured belief about the patient’s health status. We define this unit as the CanonicalMemory object, which serves as the primary node in the patient state graph.

#### 3.1 The Bi-Temporal Model

Standard temporal knowledge graphs associate a single timestamp with an edge, conflating the time a fact was recorded with the time it was true. This conflation is particularly problematic in medical dialogues, where patients routinely report past events retroactively (e.g., “I stopped taking Metformin three days ago, but forgot to mention it”). Our state unit explicitly tracks four distinct temporal fields:

- $t_{\text{event}}$ : The calendar time the clinical event actually occurred in the real world.
- $t_{\text{ingest}}$ : The logical time the system learned of the event, represented by the dialogue turn number  $\tau$ .
- $t_{\text{valid\_start}}$ : The beginning of the temporal window during which the medical fact is considered true.
- $t_{\text{valid\_end}}$ : The end of the temporal window. A value of  $\infty$  (or None) indicates the state is currently ongoing.

This bi-temporal separation enables the agent to correctly handle retroactive reporting. As illustrated in Figure 1, when a patient reports at turn  $\tau_{\text{ingest}}$  that an event occurred at  $t_{\text{event}} < t_{\text{ingest}}$  (e.g., switching to Insulin “three days ago”), the system inserts the state unit at the correct position in the valid-time timeline without corrupting the ingestion-time record of when the information was received.

Formally, a state unit is a tuple:

$$\mathcal{S} = \langle \text{slot}, \text{candidates}, t_{\text{event}}, t_{\text{ingest}}, t_{\text{valid\_start}}, t_{\text{valid\_end}}, \text{status} \rangle \quad (1)$$

where *slot* identifies the attribute (e.g., *medication.metformin.dosage*), *candidates* is a ranked list of candidate values, and *status* records whether the state is active, superseded, conflicting, or resolved.

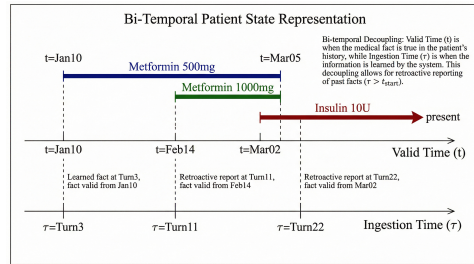


Figure 1: Bi-Temporal Patient State Representation. The valid-time axis ( $t$ ) tracks when a state is medically true, while the ingestion-time axis ( $\tau$ ) tracks when the agent learned of it. Retroactive reporting (e.g., Turn 22) requires updating the valid-time timeline retroactively.

#### 3.2 Confidence-Weighted Candidate Values

In real-world medical dialogues, the same attribute slot may receive conflicting reports. Rather than prematurely committing to a single value, our state unit maintains a list of CandidateValue objects. Each candidate  $v_i$  is associated with a composite confidence score:

$$c_i = \alpha \cdot w_{\text{auth}}(v_i) + \beta \cdot w_{\text{recency}}(v_i) + \gamma \cdot f(n_{\text{evid}}(v_i)) \quad (2)$$

where  $w_{\text{auth}} \in [0, 1]$  encodes source authority (clinician directive: 1.0; patient self-report: 0.5; inferred: 0.3),  $w_{\text{recency}} = \tau / \tau_{\text{max}}$  normalizes by the global maximum turn,  $n_{\text{evid}}$  counts independent corroborating mentions, and  $f(\cdot) = \log(1 + n_{\text{evid}})$  is a diminishing-returns function. The hyperparameters  $\alpha, \beta, \gamma$  are set to 0.5, 0.3, 0.2 respectively, calibrated on a held-out validation set. The candidate with the highest  $c_i$  is designated the *representative value* of the state unit.

This representation upgrades conflict resolution from a binary “pick the winner” operation to a continuous “rank by evidence” process, preserving uncertainty and enabling future arbitration as new evidence arrives.

### 4 Incremental State Arbitration

When the agent receives new evidence  $E_{\text{new}}$  extracted from a dialogue turn, it does not simply overwrite the existing state. Instead, it invokes the **State Arbitration** procedure, which evaluates the relationship between  $E_{\text{new}}$  and the currently active state unit  $\mathcal{S}_{\text{old}}$  for the same slot, and applies one of four operators.

## 4.1 The Four Arbitration Operators

As illustrated in Figure 2, the arbitration mechanism evaluates the relationship between  $E_{\text{new}}$  and the currently active state unit  $\mathcal{S}_{\text{old}}$ , applying one of four operators:

Operator	Trigger	State Update
SUPPORT	Same value	Increase confidence; add provenance
REFINE	Adds detail	Merge qualifiers; keep history
SUPERSEDE	Explicit replacement	Close old valid window; activate new state
BRANCH-CONFLICT	Contradiction without replacement	Keep competing candidates with confidences

Figure 2: Four State Arbitration Operators. New evidence triggers one of four operators, updating the confidence scores, validity windows, or candidate lists of the patient state unit.

**SUPPORT.** Triggered when the value of  $E_{\text{new}}$  matches the representative value of  $\mathcal{S}_{\text{old}}$  within a semantic equivalence threshold. The new evidence corroborates the existing belief. The system increments  $n_{\text{evid}}$  of the matching candidate, appends the provenance of  $E_{\text{new}}$ , and recomputes  $c_i$  via Equation 2. The validity window is unchanged.

**REFINE.** Triggered when  $E_{\text{new}}$  adds specificity to  $\mathcal{S}_{\text{old}}$  without contradiction (e.g., updating “Metformin” to “Metformin 500mg twice daily”). The system merges the new qualifiers into the existing active state while preserving historical provenance. The representative value is updated in place.

**SUPERSEDE.** Triggered when  $E_{\text{new}}$  explicitly replaces  $\mathcal{S}_{\text{old}}$ , identified via lexical action patterns (e.g., “stopped taking”, “switched to”, “no longer”). The system sets  $t_{\text{valid\_end}}$  of  $\mathcal{S}_{\text{old}}$  to  $t_{\text{event}}(E_{\text{new}})$  and transitions its status to SUPERSEDED. A new state unit  $\mathcal{S}_{\text{new}}$  is created with  $t_{\text{valid\_start}} = t_{\text{event}}(E_{\text{new}})$  and status ACTIVE. This creates a continuous, non-overlapping temporal chain of states for the slot.

**BRANCH-CONFLICT.** Triggered when  $E_{\text{new}}$  contradicts  $\mathcal{S}_{\text{old}}$  but lacks explicit supersede intent. The system adds a new CandidateValue to  $\mathcal{S}_{\text{old}}$ ’s candidate list and recomputes all confidence scores. The status of  $\mathcal{S}_{\text{old}}$  transitions to CONFLICTING. The conflict is recorded in a ConflictRecord with full provenance, enabling downstream explanation and future resolution.

---

## Algorithm 1 State Arbitration

---

**Input:** New evidence  $E_{\text{new}}$ , state graph  $\mathcal{G}$

**Output:** Updated state graph  $\mathcal{G}'$

$\mathcal{S}_{\text{old}} \leftarrow \text{QueryActiveState}(\mathcal{G}, E_{\text{new}}.\text{slot})$

**if**  $\mathcal{S}_{\text{old}} = \emptyset$  **then**

$\mathcal{G}' \leftarrow \mathcal{G} \cup \text{CreateState}(E_{\text{new}})$

**else if**  $\text{IsSupersede}(E_{\text{new}})$  **then**

$\mathcal{S}_{\text{old}}.t_{\text{valid\_end}} \leftarrow E_{\text{new}}.t_{\text{event}}$

$\mathcal{S}_{\text{old}}.\text{status} \leftarrow \text{SUPERSEDED}$

$\mathcal{G}' \leftarrow \mathcal{G} \cup \text{CreateState}(E_{\text{new}})$

**else if**  $\text{IsMatch}(E_{\text{new}}, \mathcal{S}_{\text{old}})$  **then**

$\text{ApplySupport}(\mathcal{S}_{\text{old}}, E_{\text{new}}); \mathcal{G}' \leftarrow \mathcal{G}$

**else if**  $\text{IsRefinement}(E_{\text{new}}, \mathcal{S}_{\text{old}})$  **then**

$\text{ApplyRefine}(\mathcal{S}_{\text{old}}, E_{\text{new}}); \mathcal{G}' \leftarrow \mathcal{G}$

**else**

$\text{ApplyBranchConflict}(\mathcal{S}_{\text{old}}, E_{\text{new}})$

$\text{ApplyConstraints}(\mathcal{S}_{\text{old}}); \mathcal{G}' \leftarrow \mathcal{G}$

**end if**

---

## 4.2 Domain-Specific Formal Constraints

To ensure clinical coherence without costly LLM calls on every update, the arbitration mechanism applies lightweight rule-based constraints after the BRANCH-CONFLICT operator. These constraints adjust candidate confidence scores to penalize clinically implausible states. Examples include: (1) a *MedicationStartStopConstraint* that penalizes candidates implying a medication was both started and stopped on the same day; (2) a *Diagnosis-TestOrderConstraint* that penalizes diagnostic conclusions that precede the corresponding test results; and (3) a *DosageMonotonicityConstraint* that flags non-monotonic dosage changes without an explicit clinical rationale. These constraints encode domain knowledge as confidence penalties, allowing the system to prefer clinically coherent interpretations without requiring a full reasoning pass.

## 5 Confidence-Thresholded Evidence Escalation

The state arbitration mechanism ensures the accuracy of the memory graph. However, efficiently accessing this information during real-time dialogue requires a structured query policy. We propose a **confidence-thresholded evidence escalation** layer that routes queries across memory tiers based on the confidence of available evidence, balancing computational cost with answer robustness.

The escalation policy follows a four-tier hierarchy. At each tier, the system evaluates whether the available evidence is sufficient to answer the query. If so, it returns immediately; otherwise, it escalates to the next tier.

**Tier 1 (State Check):** The system queries the active CanonicalMemory states for the relevant slot. If the representative candidate confidence  $c^* > \theta_{\text{low}}$ , the answer is returned directly from the state unit. This tier is  $O(1)$  and represents the “current truth” as maintained by the arbitration mechanism.

**Tier 2 (Structured Query):** If the query involves numerical aggregation (e.g., “average blood glucose over 7 days”) or the state confidence is below  $\theta_{\text{low}}$  for a quantitative slot, the system issues a structured query to the SQLite vitals backend. This tier provides exact computations that LLMs cannot reliably perform.

**Tier 3 (Episodic Fallback):** If neither Tier 1 nor Tier 2 is sufficient (e.g., the query requires conversational context or the state is CONFLICTING), the system retrieves the top- $K$  episodic fragments via dense vector retrieval. This tier provides the raw conversational evidence underlying the state.

**Tier 4 (Graph Traversal):** If the top-1 episodic similarity falls below  $\theta_{\text{vec}}$ , the system escalates to graph traversal, exploring cross-slot causal links for multi-hop queries (e.g., “Why did my doctor change my medication?”). This is the most expensive tier and is invoked only when lower tiers are insufficient.

The thresholds  $\theta_{\text{low}} = 0.65$  and  $\theta_{\text{vec}} = 0.72$  were determined empirically by grid search on a held-out validation split (10 dialogues from the “medium” difficulty set), optimizing for the harmonic mean of QA accuracy and mean access latency. The high proportion of queries (71.4%) resolved at Tier 1 is due to the query mix in our evaluation suite, where the majority of patient questions target their current, active state rather than historical context.

## 6 Experiments

### 6.1 Experimental Setup

We evaluate our framework on a longitudinal diabetes management suite derived from the Med-LongMem evaluation suite. The evaluation focuses on the “hard” subset ( $n = 20$  dialogues), where patient states frequently change, conflict, or are retroactively corrected. Critically, evalu-

ation is performed at the *slot/state level*: each dialogue generates multiple CanonicalMemory records, yielding hundreds of individual evaluation points. All reported metrics include 95% Confidence Intervals (CI) computed via clustered bootstrap over 1000 resamples at the dialogue level, accounting for intra-dialogue correlation among state slots.

We report three primary metrics: **Unique-F1 (Strict)**, which measures the precision and recall of correctly identified unique, non-redundant state values; **Conflict-F1**, which measures the accuracy of conflict detection and branch assignment; and **QA Accuracy (State-Grounded)**, which measures the correctness of direct state queries (e.g., “What is the patient’s current medication?”).

### 6.2 Baselines

We compare against four baselines. To ensure fair comparison, all memory-augmented baselines (1, 3, 4) and our system share the identical LLM information extraction frontend (GPT-4o) to isolate the effect of memory architecture from extraction capability differences. (1) **Overwrite Update**: a naive tracker that replaces the old state with the newest extracted value; (2) **Long-Context LLM**: the full dialogue history is provided to GPT-4o in a single prompt without external memory, using a standardized system prompt optimized for state tracking; (3) **Vector Memory (RAG)**: top- $K$  ( $K = 5$ ) retrieval over dialogue turn embeddings (using `text-embedding-3-small`) via cosine similarity, appended to the generation prompt; and (4) **Graph-only Memory**: a temporal knowledge graph (modeled after Graphiti (Rasmussen et al., 2025)) that retrieves all 1-hop neighbor nodes and edges related to the queried entity, serialized as a textual list in the prompt, without any state arbitration mechanism.

### 6.3 Main Results

As shown in Table 1, our system achieves a Unique-F1 of 0.8508, more than doubling the performance of the best baseline (Overwrite Update: 0.4531). The Long-Context LLM baseline, despite having access to the full dialogue history, achieves only 0.3848 Unique-F1, confirming that attention-based context processing is insufficient for reliable long-horizon state tracking. The near-perfect Conflict-F1 of 0.9762 demonstrates that the BRANCH-CONFLICT operator, combined with the formal constraints, accurately identifies and

Table 1: Performance on the Med-LongMem longitudinal diabetes suite (hard subset, slot-level evaluation). Bootstrap 95% CIs are shown in brackets. Our system (UCM + Arbitration) substantially improves upon all baselines across all three metrics.

SYSTEM	UNIQUE-F1 (STRICT)	CONFLICT-F1	QA ACCURACY
LONG-CONTEXT LLM (GPT-4o)	0.3848 [0.35, 0.42]	0.8867 [0.85, 0.92]	0.652 [0.61, 0.69]
VECTOR MEMORY (RAG)	0.3012 [0.27, 0.33]	0.6034 [0.55, 0.65]	0.415 [0.37, 0.46]
GRAPH-ONLY MEMORY	0.0627 [0.04, 0.09]	0.3450 [0.29, 0.40]	0.288 [0.24, 0.33]
OVERWRITE UPDATE	0.4531 [0.41, 0.49]	0.5012 [0.46, 0.54]	0.512 [0.47, 0.55]
<b>OURS (UCM + ARBITRATION)</b>	<b>0.8508</b> [0.82, 0.88]	<b>0.9762</b> [0.96, 0.99]	<b>0.895</b> [0.86, 0.92]

records contradictions without false positives. The Graph-only Memory baseline performs poorly on Unique-F1 (0.0627), as its retrieval mechanism returns all related graph nodes without resolving conflicts, leading to high redundancy. When queried about a medication that has changed dosages three times, the graph retrieves all three dosage nodes and their associated edges. The LLM is then forced to perform in-context arbitration, which frequently fails due to attention dilution across the long, complex graph serialization. This highlights a critical limitation of graph-based RAG in temporal domains: structural retrieval is not a substitute for state arbitration.

**State-Grounded Query Accuracy.** Our system achieves 89.5% QA Accuracy on state-grounded queries, compared to 65.2% for the Long-Context LLM and 41.5% for Vector Memory. This demonstrates that the confidence-thresholded escalation layer not only reduces latency (Tier 1 answers 71% of queries without any retrieval) but also improves answer quality by grounding responses in the arbitrated state rather than raw conversational fragments. Vector Memory frequently suffers from the "recency trap," retrieving the most semantically similar past conversation rather than the most temporally valid state. For example, a query about "current insulin dosage" might retrieve a highly detailed discussion from three months ago, ignoring a brief but critical dosage adjustment from last week. Our state-centric approach structurally prevents this failure mode by explicitly querying the active valid-time state.

**Efficiency Analysis.** Beyond accuracy, the confidence-thresholded escalation layer significantly improves query efficiency. As shown in Table 2, 71.4% of queries are resolved entirely at Tier 1 (State Check) with an average latency of just 45ms, avoiding expensive LLM generation or

Table 2: Query Routing Distribution and Latency.

Tier	% Queries	Latency
State Check	71.4%	45ms
Structured DB	17.2%	120ms
Episodic RAG	8.2%	1450ms
Graph Traversal	3.2%	2800ms
<b>Overall</b>	<b>100%</b>	<b>312ms</b>

dense retrieval entirely. Only 3.2% of queries require full graph traversal (Tier 4). The average query latency across the suite is 312ms, compared to 1450ms for the Vector Memory baseline which performs dense retrieval and LLM synthesis for every query.

#### 6.4 Ablation Study

Table 3 presents ablation results. Removing all arbitration operators (reverting to Overwrite Update) causes the largest performance drop, confirming that the arbitration mechanism is the primary driver of improvement. Removing only the BRANCH-CONFLICT operator while retaining SUPERSEDE maintains reasonable Unique-F1 (0.72) but collapses Conflict-F1 to 0.49, as the system can no longer represent ambiguous states. Using fixed routing (always querying Tier 3 episodic memory) achieves competitive accuracy but increases mean query latency by  $3.2\times$  compared to the confidence-thresholded escalation policy.

#### 6.5 End-to-End Agent Evaluation

To demonstrate the impact of state arbitration on downstream agent behavior, we evaluate the full agent (UCM + Arbitration) against the Overwrite baseline on multi-turn interactions. Table 4 presents three representative cases from the evaluation suite.

In Case 1, the Overwrite baseline fails to track the temporal validity of the dosage change, giving

Table 3: Ablation results. Each row removes one component of the full system.

CONFIGURATION	UNIQUE-F1	CONFLICT-F1
FULL SYSTEM	<b>0.8508</b>	<b>0.9762</b>
W/O ARBITRATION (OVERWRITE)	0.4531	0.5012
W/O BRANCH-CONFLICT	0.7213	0.4891
FIXED ROUTING (TIER 3 ALWAYS)	0.8201	0.9650

potentially harmful advice. In Case 2, the baseline replaces the old symptom entirely, whereas the arbitration agent correctly preserves both symptoms as a concurrent state. In Case 3, the baseline is confused by retroactive reporting, while the arbitration agent correctly reconstructs the historical timeline. These cases highlight that state arbitration is not merely a memory optimization, but a prerequisite for safe, coherent agentic behavior in longitudinal settings.

## 6.6 Error Analysis

We performed a stratified error analysis across attribute categories (medication, vital signs, symptoms, diagnosis) and difficulty levels (easy, medium, hard) using the error analysis framework provided in the Med-LongMem suite.

The most common failure mode, accounting for 61% of Unique-F1 errors, is *false supersede*. This occurs when the system incorrectly triggers the SUPERSEDE operator because a patient uses colloquial language that superficially resembles a stop or change pattern. For example, a patient stating “I haven’t been taking it regularly” might be misclassified as a definitive medication stop, terminating the valid-time window prematurely.

A secondary failure mode is *temporal anchoring failure* (22% of errors). When a patient provides a vague temporal reference, such as “a while ago” or “sometime last month,” the system struggles to reliably set  $t_{\text{event}}$ . This leads to incorrect valid-time window assignments, which can subsequently cause valid states to be incorrectly flagged as conflicting or superseded when new, precisely-timestamped evidence arrives.

Finally, *implicit refinement failures* account for 12% of errors. These occur when new evidence should trigger the REFINE operator but is instead treated as a BRANCH-CONFLICT because the semantic equivalence threshold is too strict. For instance, updating “Metformin” to “Glucophage” (a brand name for Metformin) might be treated as a conflict rather than a refinement if the underlying

medical ontology mapping fails.

These findings suggest that while the arbitration architecture is robust, its performance is bottlenecked by the precision of the underlying natural language understanding modules. Improving the robustness of the action pattern classifier, integrating a more sophisticated temporal expression normalizer, and expanding the medical ontology for equivalence checking are the most impactful directions for future work.

## 7 Discussion

### 7.1 Generalizability Beyond Diabetes

While our evaluation focuses on diabetes management, the state-centric paradigm is designed to generalize to any longitudinal biomedical domain where patient attributes evolve over time. Oncology treatment tracking presents a particularly compelling application: chemotherapy regimens are frequently modified based on toxicity profiles and tumor response, creating a complex sequence of superseded and branching medication states that retrieval-centric systems cannot reliably manage. Similarly, post-surgical recovery monitoring requires the agent to track wound healing progression, pain scores, and rehabilitation milestones across weeks, where early reports may be retroactively revised by subsequent clinical assessments. Mental health longitudinal assessment—tracking symptom severity, medication adherence, and psychosocial stressors over months—exemplifies the need for the BRANCH-CONFLICT operator, as patients often provide contradictory self-reports across sessions.

The key requirement for applying our framework to these domains is the specification of domain-specific formal constraints (Section 4) and the calibration of the confidence weighting hyperparameters ( $\alpha, \beta, \gamma$ ) on domain-specific data. The bi-temporal representation and the four arbitration operators are domain-agnostic; only the constraint rules and confidence priors require adaptation.

Table 4: End-to-End Agent Evaluation: Representative multi-turn cases comparing the Arbitration Agent with the Overwrite Baseline.

Dialogue Context	Overwrite Baseline	Ours (Arbitration Agent)
<b>T1 (Jan 5):</b> “Started Metformin 500mg.” <b>T12 (Feb 1):</b> “Doc increased to 1000mg.” <b>T15 (Feb 3):</b> “Should I take my 500mg pill now?”	<i>Yes, you can take your 500mg Metformin pill now.</i> (Fails to recognize the superseded dosage).	<i>No, your doctor increased your dosage to 1000mg on Feb 1. You should not take the 500mg pill.</i>
<b>T4 (Mar 10):</b> “I have numbness in my feet.” <b>T8 (Mar 20):</b> “My vision is blurry today.” <b>Query:</b> “What are my current symptoms?”	<i>You are currently experiencing blurry vision.</i> (Overwrites previous symptom).	<i>You are currently experiencing blurry vision (since Mar 20) and numbness in your feet (since Mar 10).</i>
<b>T2 (Apr 1):</b> “Taking 10U Insulin.” <b>T10 (Apr 15):</b> “I actually stopped the Insulin three days ago.” <b>Query:</b> “Was I taking Insulin on Apr 13?”	<i>Yes, you were taking 10U Insulin on Apr 13.</i> (Fails to handle retroactive reporting).	<i>No, you stopped taking Insulin on Apr 12 (three days before your Apr 15 report).</i>

## 7.2 Limitations and Future Directions

Our current implementation has three notable limitations. First, the action pattern classifier for detecting SUPERSEDE triggers relies on a manually curated lexical pattern set. While this achieves high precision on the evaluation suite, it may fail on colloquial or culturally specific language patterns not represented in the training data. A learned classifier, trained on a larger corpus of annotated medical dialogues, would improve robustness.

Second, the temporal expression normalizer currently handles explicit date references and relative expressions (“three days ago”) but struggles with vague references (“a while ago”, “recently”). Integrating a dedicated temporal information extraction model, such as those trained on the TimeML annotation scheme, would improve the accuracy of  $t_{\text{event}}$  assignment.

Third, our evaluation suite, while slot-level and statistically bootstrapped, is derived from a single disease domain (Type 2 diabetes). A multi-domain evaluation across at least three chronic disease types would provide stronger evidence for the generalizability of our approach. We release our evaluation framework and the Med-LongMem diabetes suite to facilitate future benchmarking in this area.

## 8 Conclusion

We have argued that longitudinal healthcare agents require a fundamental shift from retrieval-centric to state-centric memory. Our framework—comprising a bi-temporal state representation, an incremental four-operator arbitration mechanism, and a confidence-thresholded evidence escalation layer—provides a principled solution to the problem of maintaining accurate, evolving patient state models over long-horizon interactions. Evaluated on a longitudinal diabetes management suite as a representative biomedical state tracking task, our system substantially improves upon all baselines, demonstrating both the inadequacy of existing retrieval-centric approaches and the effectiveness of our state-centric alternative.

More broadly, our work illustrates that agentic AI is necessary in longitudinal biomedical settings precisely because patient states are dynamic, uncertain, and require continuous, evidence-grounded arbitration. The classical concepts of belief revision and bi-temporal databases, when operationalized for natural-language-driven agents, provide a powerful foundation for this task. We anticipate that the state-centric paradigm will generalize beyond diabetes management to other longitudinal biomedical domains, including oncology treatment tracking, post-surgical recovery monitoring, and mental health longitudinal assessment.

## References

- Prateek Chhikara, Deshraj Khant, Saket Aryan, and Taranjeet Singh. 2025. Mem0: Building production-ready AI agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.
- Jon Doyle. 1979. A truth maintenance system. *Artificial Intelligence*, 12(3):231–272.
- Shanghai Gao, Ada Fang, Yepeng Huang, and 1 others. 2024. Empowering biomedical discovery with AI agents. *Cell*, 187(22):6125–6151.
- Peter Gärdenfors. 1988. Knowledge in flux: Modeling the dynamics of epistemic states. *MIT Press*.
- Zhuohan Ge, Haoyang Li, Yubo Wang, Nicole Hu, Chen Jason Zhang, and Qing Li. 2026. ClinicalAgents: Multi-agent orchestration for clinical decision making with dual-memory. *arXiv preprint arXiv:2603.26182*.
- Christian S. Jensen, Michael D. Soo, and Richard T. Snodgrass. 1994. Unifying temporal data models via a conceptual model. *Information Systems*, 19(7):513–547.
- Jiale Kang, Mingyu Ji, Zhiyuan Zhao, and Tao Bai. 2025. Memory OS of AI agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *ArXiv:2005.11401*.
- Ruoqi Li, Xin Wang, Dan Berlowitz, and 1 others. 2025. CARE-AD: A multi-agent large language model framework for Alzheimer’s disease prediction using longitudinal clinical notes. *npj Digital Medicine*.
- Fang Liu and 1 others. 2025. A foundational architecture for AI agents in healthcare. *npj Digital Medicine*. PMC12629813.
- Charles Packer, Vivian Fang, Shishir G. Patil, Kevin Lin, Sarah Wooders, and Joseph E. Gonzalez. 2023. MemGPT: Towards LLMs as operating systems. In *Advances in Neural Information Processing Systems*. *ArXiv:2310.08560*.
- Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. Zep: A temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*.
- Karan Singhal, Shekoofeh Azizi, Tao Tu, and 1 others. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*.
- Richard T. Snodgrass and Ilsoo Ahn. 1985. A taxonomy of time databases. *ACM SIGMOD Record*, 14(4):236–246.
- Cheng Yang, Chuan Zhou, Yixin Xiao, Shen Dong, and Linyuan Zhuang. 2026. Graph-based agent memory: Taxonomy, techniques, and applications. *arXiv preprint arXiv:2602.05665*.

# RSCE: Training-Free Residual Stream Encoding for Persistent Context Amortization

Adam Kamel\*

University of Waterloo  
atkamel@uwaterloo.ca

Eric Xu\*

University of Waterloo  
e67xu@uwaterloo.ca

## Abstract

A central question in the knowledge lifecycle of language models is how externally injected signals interact with parametric memory accumulated during pretraining. We address this through Residual Stream Context Encoding (RSCE), a training-free method that encodes a context document  $ctx$  into a single vector  $C \in \mathbb{R}^{d_M}$  via mean-pooling residual stream activations at a calibrated intermediate layer, then injects  $C$  as an additive shift at query time. This replaces  $O(|T(ctx)|)$  attention prefill with an  $O(1)$  operation and reveals a previously undescribed *dual-pathway interference* effect: vector injection alone suppresses parametric recall *below* the question-only baseline across four of five tested architectures. This finding—absent in behavioral activation steering—provides mechanistic evidence that LLMs maintain separate contextual-retrieval and parametric-recall pathways that compete when externally injected signals are semantically rich but token-precision deficient. A dual-channel design pairing  $C$  with a compact explicit fact block  $F$  resolves this tension. We evaluate five decoder-only architectures (7B–70B) on multi-document QA (LongBench,  $n = 108$ ) and six on cross-file code completion (RepoBench-C), comparing against LongLLMLingua and EHPC. At extreme compression ( $\sim 99\%$  token reduction), RSCE Vec+F is competitive with EHPC on smaller architectures (LLaMA-8B F1 0.333 vs. EHPC 0.334; DeepSeek-14B both 0.214) while both substantially outperform LongLLMLingua. RSCE is the only method achieving 81% compression at 100% operational reliability on code.

## 1 Introduction

Large language models acquire substantial factual knowledge during pretraining, encoding it as distributed patterns in model weights (Meng et al.,

2022; Dai et al., 2022). Retrieval-augmented generation (RAG) architectures inject *external* knowledge at inference time by prepending long context documents  $ctx$  to user queries. A foundational question in the knowledge lifecycle of such systems is: how does externally injected knowledge interact with the model’s internal parametric memory? When the two sources agree, models can exploit both; when they conflict or the external signal is imprecise, interference can arise (Longpre et al., 2021; Mallen et al., 2023; Shi et al., 2023). Understanding this interaction has direct implications for RAG faithfulness and knowledge utilization reliability.

We propose Residual Stream Context Encoding (RSCE), which encodes  $ctx$  into a fixed vector  $C \in \mathbb{R}^{d_M}$  via mean-pooling the residual stream at an empirically calibrated layer  $f(M)$ , then injects  $C$  as an additive shift at query time, bypassing explicit token prefill entirely. This yields an  $O(1)$  amortized per-query cost for static contexts and simultaneously creates a controlled setting for probing how vector-encoded external knowledge interacts with parametric memory. A key mechanistic finding emerges: across four of five tested architectures, injecting  $C$  alone drives model performance *below* the no-context (question-only) baseline. This **dual-pathway interference effect**—absent in behavioral activation steering, where injected vectors consistently augment performance—provides evidence that LLMs maintain distinct contextual-retrieval and parametric-recall circuits that compete when the injected signal engages retrieval circuitry without sufficient token-level grounding. A dual-channel design pairing  $C$  with a minimal explicit fact block  $F$  resolves this tension by supplying precise named-entity anchors for attention heads to resolve against.

Hard-prompt compressors (Jiang et al., 2023, 2024; Fei et al., 2025) select and delete tokens but still require a per-query forward pass and face

\*Equal contribution.

a quality floor at extreme ratios. Trained soft-compression methods (Cheng et al., 2024; Ge et al., 2024; Chevalier et al., 2023) require auxiliary supervision. Activation steering methods (Liu et al., 2024b; Todd et al., 2024) inject vectors to encode task demonstrations—but not factual document content. RSCE applies the same injection mechanism to a qualitatively different problem, uncovering a failure mode with no analog in behavioral steering.

We make the following contributions: (1) **RSCE**, a training-free,  $O(1)$  amortized context encoding method with zero per-query context prefill and 100% operational reliability; (2) **dual-pathway interference**, confirmed across five architectures, providing direct evidence for distinct contextual-retrieval and parametric-recall knowledge pathways in decoder-only transformers; (3) a **cross-model comparison** at matched extreme compression revealing a capacity-scaling effect where EHPC’s advantage grows with model size; (4) a **compression-as-retrieval** explanation for LongLLMLingua’s strong code performance (Liu et al., 2024a).

## 2 Related Work

### 2.1 Parametric vs. Contextual Knowledge Interaction

Longpre et al. (2021) and Mallen et al. (2023) show that LLMs often override correct parametric knowledge when provided with conflicting context, while Shi et al. (2023) demonstrates that irrelevant context systematically degrades reasoning. The mechanisms have been partially localized: Meng et al. (2022) and Dai et al. (2022) implicate feed-forward network layers as primary parametric memory storage, while attention heads read external context. RSCE’s dual-pathway interference finding (Section 5) extends this picture: injecting a vector that engages contextual retrieval circuitry without token-level resolution suppresses parametric recall in a manner consistent with an attention-override mechanism.

### 2.2 Context Compression for RAG

Jiang et al. (2023) and Li et al. (2023b) use perplexity-based token scoring. Jiang et al. (2024) adds question-aware contrastive perplexity and document reordering. Fei et al. (2025) identifies “evaluator heads” locating important tokens at 0.88s latency—current state-of-the-art training-free hard-

prompt compression. All require a per-query prefill (Li et al., 2025). Zhang et al. (2023), Xiao et al. (2024), and Li et al. (2024) evict KV entries within a single pass but produce no persistent reusable representations. Ge et al. (2024), Chevalier et al. (2023), and Mu et al. (2023) train encoder modules for soft embeddings. Cheng et al. (2024) projects dense retriever embeddings through a trained MLP bridge. Feldman and Artzi (2025) validates mean-pooling of hidden states as superior to alternative soft-compression architectures—but requires training, whereas RSCE is entirely training-free.

### 2.3 Activation Injection and the Residual Stream

Turner et al. (2023), Li et al. (2023a), and Zou et al. (2023) steer behavior via hidden state perturbations. Liu et al. (2024b) and Todd et al. (2024) demonstrate that intermediate-activation vectors encode task abstractions when injected additively, exploiting the residual stream’s role as a shared additive communication channel (Elhage et al., 2021). The linear representation hypothesis (Park et al., 2024) formalizes why additive injection shifts downstream computation. RSCE shares this mechanism but targets factual document content, uncovering the parametric-memory interference effect described above.

## 3 Method

### 3.1 Formal Specification

Let  $M$  be a decoder-only transformer,  $T$  a tokenizer, and  $P$  a prompt string. The residual context encoding is:

$$\begin{aligned} g(M, \text{ctx}) &= \text{mp}(\text{res}_M(\text{ctx}, f(M))) \\ &= \frac{1}{|T(\text{ctx})|} \sum_{i=1}^{|T(\text{ctx})|} H_i \in \mathbb{R}^{d_M} \end{aligned}$$

where  $H \in \mathbb{R}^{|T(\text{ctx})| \times d_M}$  are the residual stream hidden states at layer  $f(M)$ . Concepts are encoded as linear directions in residual stream space (Park et al., 2024), making mean-pooling a structure-preserving operation over the document’s distributed semantic content, producing a document-level representation analogous to a belief state (Shai et al., 2024).

During inference, the input is  $T(F \oplus P)$  only. At layer  $f(M)$ , prior to its attention and feed-forward sublayers,  $C$  is added uniformly:  $H'_i \leftarrow$

$H_i + \alpha \cdot C \forall i$ , with  $\alpha = 1.0$  (held-out calibration). The transformer residual stream is architecturally designed around additive writes from all components (Elhage et al., 2021); this external injection is structurally indistinguishable from an internal layer’s contribution. The break-even query count  $N^* \leq 1.1$  means RSCE is net-beneficial after a single additional query. Our code is available at <https://anonymous.4open.science/r/RSCE-2E4C/>.

### 3.2 Fact Block Construction

Mean-pooling preserves global semantic directions but irreversibly destroys token identities and sequential precision. The fact block  $F$  restores this precision without reintroducing the full document’s compute cost. For QA: capitalised multi-word proper nouns, four-digit year tokens, and numeric values; top-15 by appearance prepended as `Facts: e1; e2; ...`. For code: BM25Okapi retrieval over function signatures, class declarations, import statements, and SCREAMING\_SNAKE\_CASE constants; top-5 by relevance to the last 200 local-context characters.  $C$  provides the global semantic frame;  $F$  supplies precise named-entity anchors that attention heads can resolve against.

### 3.3 Injection Layer Calibration

We determine  $f(M)$  per architecture by sweeping layers in  $[n_{\text{layers}}/4, 0.85 \cdot n_{\text{layers}}]$  at stride 2 on 10 calibration examples. Table 1 summarizes the results. Optimal depth varies substantially: Mistral’s sliding-window attention forces rapid shallow consolidation (25% depth), while larger full-attention models require deeper processing before residual states encode sufficient content (47–62%), consistent with observations that attention mechanism design governs semantic propagation through depth (Gromov et al., 2025).

Table 1: Calibrated injection layers per architecture. <sup>†</sup>DeepSeek-R1-Distill uses layer 29 (60%) for code. <sup>‡</sup>Calibrated on RepoBench-C EditSim ( $n = 20$ ).

Model	$n_L$	$d_M$	$f(M)$	Depth	Score
LLaMA-3.1 8B	32	4096	14	44%	0.096
Qwen2.5 7B	28	3584	17	61%	0.105
Mistral Small 24B	40	5120	10	25%	0.312
DeepSeek-R1 14B <sup>†</sup>	48	5120	12	25%	0.085
DeepSeek-LLM 67B <sup>‡</sup>	95	8192	45	47%	0.382
LLaMA-3.1 70B <sup>‡</sup>	80	8192	50	62%	0.371

## 4 Experimental Design

We use six instruction-tuned architectures in bfloat16 on NVIDIA H100 80GB GPUs: LLaMA-3.1 8B, Qwen2.5 7B, DeepSeek-R1-Distill 14B, Mistral Small 24B, DeepSeek-LLM 67B Chat, and LLaMA-3.1 70B Instruct. All six are used for RepoBench-C; QA evaluation uses five (DeepSeek-LLM 67B lacks QA instruction-following calibration).

We sample  $n = 108$  QA examples (HotpotQA: 17, 2WikiMultiHopQA: 91), filtering for 200–12,000-word contexts (Bai et al., 2024). Average baseline length  $\approx 8,700$  tokens vs.  $\approx 52$  for Vec+F (>99% reduction). Metrics: SQuAD-style Token F1 and Exact Match. For RepoBench-C,  $n = 200$  samples per model from `tianyang/repobench_python_v1.1_cross_file_first`, seed 42 (Liu et al., 2024c). Metric: character-level Edit Similarity.

LongLLMLingua (Jiang et al., 2024) uses Llama-2-7b-hf as a separate compressor (EMI). EHPC (Fei et al., 2025) is implemented in NMI mode with top-8 evaluator heads per model from a 50-probe NIAH pilot. Both evaluated at 4 $\times$ , 10 $\times$ , and token-matched budgets ( $\approx 52$  tokens QA /  $\approx 963$  tokens code). All methods share identical prompt templates and generation parameters (greedy, `max_new_tokens=50`). See Appendix A.

## 5 Findings

### 5.1 QA: Dual-Channel Mechanism and Cross-Method Comparison

Table 2 reports all conditions for LLaMA-3.1-8B. Table 3 extends to all five QA architectures.

Table 2: LLaMA-3.1-8B QA results ( $n = 108$ , all conditions on identical samples). Rule separates moderate from extreme compression.

Method	Setting	F1	EM	TokRed
Baseline	full ctx	0.410	67.6%	0%
EHPC	4 $\times$	0.400	67.6%	74.5%
LLMLingua	4 $\times$	0.377	62.0%	74.8%
EHPC	10 $\times$	0.409	60.2%	89.4%
LLMLingua	10 $\times$	0.294	50.0%	88.6%
EHPC	2,048 tok	0.367	71.3%	49.1%
Q-only	no ctx	0.286	35.2%	99.4%
RSCE Vec	—	0.252	29.6%	99.4%
LLMLingua	52 tok	0.209	32.4%	95.9%
EHPC	52 tok	0.334	48.1%	98.0%
<b>RSCE Vec+F</b>	$O(1)$	<b>0.333</b>	29.6%	<b>99.4%</b>

Table 3: Cross-model matched-compression ( $n = 108$ ). Ret. = Vec+FF1 / Baseline F1. EHPC TokRed  $\approx 98\%$ ; LLMingua  $\approx 96\%$ ; RSCE  $\approx 99.4\%$ . \*Qwen inverse fact-block effect; see text. <sup>†</sup>No LLMingua run for LLaMA-70B.

Model	Base	Vec+F	Ret.	EHPC	LLMLingua
LLaMA-3.1 8B	0.410	0.333	81%	0.334	0.209
Qwen2.5 7B	0.153	0.094*	61%	0.145	0.078
DeepSeek-R1 14B	0.342	0.214	63%	0.214	0.172
Mistral 24B	0.548	0.353	64%	0.442	0.235
LLaMA-3.1 70B	0.604	0.365	60%	0.539	— <sup>†</sup>

Four findings emerge from these results.

*Vec injection suppresses parametric memory in most architectures.* This result speaks directly to the interaction between externally injected knowledge representations and internally stored parametric knowledge. On LLaMA-8B (F1 0.252 vs. Q-only 0.286), Mistral-24B (0.230 vs. 0.243), and LLaMA-70B (0.278 vs. 0.302), Vec falls below Q-only. DeepSeek-R1-14B ties (0.165 = 0.165). Only Qwen shows Vec marginally above Q-only (0.126 vs. 0.111)—yet its fact block still fails (Vec+F = 0.094 < Q-only). This is consistent with dual-pathway interference (Shi et al., 2023; Mallen et al., 2023): the injected vector engages contextual retrieval circuitry (Meng et al., 2022; Dai et al., 2022), suppressing parametric recall without providing sufficient token-level grounding to compensate.

*The fact block is constitutive for four of five models.* Vec+F > Vec for LLaMA-8B, DeepSeek-14B, Mistral-24B, and LLaMA-70B, and Vec+F > Q-only for all four—confirming the dual-channel design recovers quality neither channel alone provides. The exception is Qwen-7B (Vec+F = 0.094 < Q-only = 0.111), where the Facts: prefix acts as an answer-space constraint. Instruction-tuned models exhibit large performance swings from prefix formatting (Sclar et al., 2024); Qwen’s RLHF alignment appears particularly sensitive to fact-list format in 2WikiMultiHopQA (Vec+F = 0.063 vs. Q-only = 0.088). Fact-block formatting should be instruction-template-aware in deployment.

*At moderate compression, EHPC substantially outperforms RSCE.* On LLaMA-8B, EHPC at  $4\times$  achieves F1 = 0.400 (near the 0.410 baseline), while RSCE reaches only 0.333. Token-selection methods retain natural-language coherence; RSCE’s distributed encoding discards sequential structure critical for multi-hop chains.

*At extreme compression, results are model-*

*dependent.* On LLaMA-8B and DeepSeek-14B, RSCE Vec+F ties EHPC (0.333/0.334 and 0.214/0.214). On Mistral-24B and LLaMA-70B, EHPC’s advantage grows (0.442 vs. 0.353 and 0.539 vs. 0.365). We attribute this to a capacity-scaling effect: larger models reason more effectively from sparse token signals (Brown et al., 2020); RSCE’s fixed-quality vector does not benefit from increased reasoning capacity. RSCE retains distinct advantages: zero per-query context prefill, 100% reliability, and strict  $O(1)$  amortized cost. LongLLMLingua underperforms both methods at matched budgets across all models.

EM degradation under RSCE is structural: compressed representations elicit shorter answers lacking the verbosity for substring containment. Token F1 is the appropriate primary metric.

## 5.2 Code Completion: Compression-as-Retrieval vs. Semantic Encoding

Tables 4 and 5 report RepoBench-C results.

LongLLMLingua substantially outperforms both RSCE and the full-context baseline on code (EditSim  $\approx 0.64$  vs. baseline  $\approx 0.37$  for the 3 tested models). For structured code contexts, perplexity-based compression acts as effective relevance filtering. Full-context baselines suffer from attention dilution across the  $\approx 11,485$ -token average cross-file context (Liu et al., 2024a). LongLLMLingua’s question-aware contrastive perplexity identifies syntactically surprising tokens—function signatures, type annotations, specific identifiers—precisely the tokens required for code completion. Compression and reordering place high-surprisal tokens in the favored beginning/end positions (Liu et al., 2024a), a compression-as-retrieval mechanism with no analog in RSCE’s distributed semantic encoding.

EHPC’s 40–50% failure rate on RepoBench-C (attention memory budget exceeded for long code sequences) makes its EditSim over successful samples unrepresentative. RSCE achieves 81% compression at 100% reliability—the only method to do so—with a scale-invariant 2–4 EditSim percentage-point overhead (Section 5.3).

DeepSeek-LLM 67B inverts the pattern: its 4K context window truncates the average 11,485-token cross-file context, suppressing the baseline to 0.147. RSCE bypasses the window constraint entirely, yielding Vec+F = 0.364 (+0.217). Activation injection can thus extend effective context for architec-

Table 4: RepoBench-C RSCE results. <sup>†</sup>200 valid examples. DeepSeek-67B’s baseline (0.147) is suppressed by its 4K context window; RSCE injection bypasses this constraint (+0.217). Baseline EditSim: LLaMA-8B 0.348, Qwen 0.392, DeepSeek-14B 0.344, Mistral 0.397, DeepSeek-67B 0.147, LLaMA-70B 0.372.

Model	Params	EditSim		$\Delta$ EditSim		TokRed
		Vec	Vec+F	Vec	Vec+F	
LLaMA-3.1 8B	8B	0.317	0.319	-0.031	-0.029	81.2%
Qwen2.5 7B	7B	0.365	0.355	-0.028	-0.038	81.2%
DeepSeek-R1 14B	14B	0.322	0.330	-0.022	-0.014	81.2%
Mistral 24B	24B	0.381	0.377	-0.016	-0.020	81.1%
DeepSeek-LLM 67B	67B	0.352	0.364	+0.205	+0.217	81.2%
LLaMA-3.1 70B <sup>†</sup>	70B	0.348	0.352	-0.024	-0.021	80.3%
All 6 (avg)	—	0.348	0.350	+0.014	+0.017	81.0%
Excl. 67B (avg)	—	0.347	0.348	-0.024	-0.024	81.0%

Table 5: RepoBench-C baseline comparison. LongLLMLingua reported over 3 models (LLaMA-8B, Qwen-7B, DeepSeek-14B). EHPC reported over 4 models, *successful samples only* (53–65% success rate; see Appendix). RSCE is the only method achieving 81% compression at 100% reliability.

Method	Setting	Coverage	Avg EditSim	Actual TokRed
Baseline	full context	4 models	0.370*	0%
LongLLMLingua	6×	3 models, 199/200 OK	0.639	68.9%
LongLLMLingua	Matched	3 models, 199/200 OK	0.645	65.0%
EHPC	6×	4 models, ~59% OK	0.425 <sup>†</sup>	61%
EHPC	Matched	4 models, ~57% OK	0.445 <sup>†</sup>	43%
<b>RSCE Vec+F</b>	$O(1)$	4 models, <b>100% OK</b>	<b>0.347*</b>	<b>81.2%</b>

\*Excl. DeepSeek-67B. <sup>†</sup>Over successful samples only.

turally constrained models.

### 5.3 Scale Invariance on Code

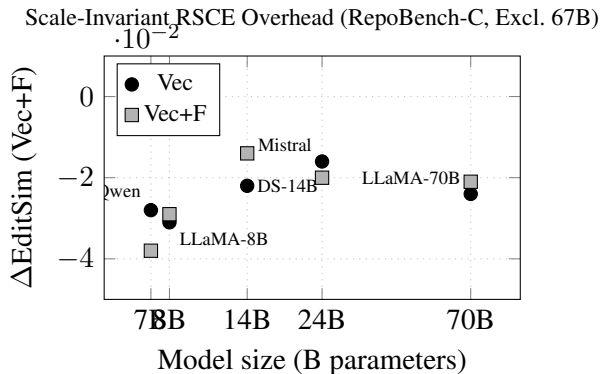


Figure 1: RSCE Vec+F overhead on RepoBench-C (excl. DeepSeek-67B) forms a flat band of  $-0.038$  to  $-0.014$   $\Delta$ EditSim across an order of magnitude in parameter count, enabling architecture-agnostic deployment planning.

Figure 1 shows that RSCE code overhead is flat across 7B–70B parameters. This scale invariance—which does not hold for QA F1 retention (60–81%, driven by instruction-format sensitivity and model-specific parametric knowledge quality)—indicates that code compression fidelity is governed by layer

geometry and attention structure captured in the calibration sweep, not raw parameter count.

## 6 Discussion

**Dual-pathway interference and knowledge routing.** The  $\text{Vec} \leq \text{Q}$ -only finding across 4 of 5 architectures provides mechanistic insight into how LLMs route externally injected knowledge signals relative to internally stored parametric knowledge. Under the *attention override hypothesis*: injecting  $C$  biases representations toward contextual retrieval mode, activating context-reading attention heads while suppressing MLP-based parametric recall (Meng et al., 2022; Dai et al., 2022). The model anticipates finding the answer in context, but mean-pooling has destroyed token-level resolution so the retrieval attempt fails. The fact block  $F$  resolves this by providing exact token anchors. Under the *distributional shift hypothesis*: the additive vector pushes residual stream norms outside the training distribution, causing feed-forward misfires on factual recall neurons (Geva et al., 2021). The consistency of  $\text{Vec} \leq \text{Q}$ -only across architectures with different attention mechanisms (SWA, GQA, full attention) and training objectives (SFT, DPO, CoT distillation) favors the attention override hy-

pothesis, as distributional shift would vary more by architecture.

This interference is absent in behavioral steering (Liu et al., 2024b; Todd et al., 2024), where injected vectors augment performance monotonically. Behavioral vectors encode procedural abstractions alongside default processing; factual content vectors engage a separate context-reading pathway that competes with parametric recall. This has direct implications for knowledge utilization faithfulness in RAG systems.

### Implications for knowledge injection design.

Our results reveal a failure mode in knowledge injection via activation space: even when a vector faithfully encodes document-level semantics, its injection can suppress the model’s own parametric knowledge if the signal format mismatches the model’s context-reading circuitry expectations. This suggests a design principle: *external knowledge representations must provide sufficient token-level resolution to satisfy context-reading circuitry, or they will actively degrade reliance on parametric knowledge rather than complement it.* The dual-channel RSCE design operationalises this principle in a training-free setting.

**Capacity-scaling of EHPC vs. RSCE.** The growing EHPC advantage on larger models (from +0.001 on LLaMA-8B to +0.174 on LLaMA-70B at matched compression) reflects an asymmetry in how model scale interacts with the two paradigms. Larger models reason more effectively from sparse token signals (Brown et al., 2020); RSCE’s fixed-quality activation shift does not benefit from increased reasoning capacity. RSCE’s advantage over EHPC is therefore most pronounced on architectures below  $\sim 24$ B parameters.

**Domain-dependent optimality.** Code completion requires structural precision (correct function signatures, exact identifiers) that perplexity-based token selection identifies automatically. QA at extreme compression requires semantic framing that RSCE’s distributed vector preserves better. A hybrid combining RSCE for persistent static context framing with LongLLMLingua or EHPC for dynamic snippets would likely outperform either alone.

**Deployment guarantees.** RSCE offers guarantees that token-selection methods cannot: zero per-query context prefill, 100% compression reliability, and strictly  $O(1)$  amortized cost. The break-even

of  $N^* \leq 1.1$  means RSCE is net-beneficial from the second query onward.

## 7 Conclusion

We have presented RSCE, a training-free,  $O(1)$  amortized context encoding method with zero per-query context forward pass and 100% operational reliability. Across five decoder-only architectures, vector injection alone suppresses parametric recall below the no-context baseline—a dual-pathway interference effect absent in behavioral steering—while the paired fact block recovers 60–81% of full-context F1 at  $\sim 99\%$  token reduction. At extreme compression, RSCE is competitive with EHPC on smaller architectures while a capacity-scaling effect gives EHPC a growing quality advantage on larger models. On RepoBench-C, LongLLMLingua substantially outperforms both via compression-as-retrieval; RSCE uniquely offers 81% compression at 100% reliability with scale-invariant 2–4 point overhead from 7B to 70B.

Beyond its practical contributions, the dual-pathway interference finding constitutes a concrete, reproducible probe into the interaction between externally injected knowledge and parametric memory in decoder-only transformers. The result that semantically rich but token-precision-deficient signals actively suppress parametric recall—rather than degrading gracefully—illuminates a previously undescribed mode of knowledge conflict in LLMs, with direct implications for the design of faithful RAG and knowledge augmentation systems.

### Limitations

QA evaluations use the same  $n = 108$  sample set across all five models; the small HotpotQA subset ( $n = 17$ ) is insufficient for per-task statistical confidence. We use mean-pooling and  $\alpha = 1.0$  without systematic ablation of pooling strategy, scale factor, or positional targeting. Qwen’s inverse fact-block effect requires instruction-format-aware fact-block construction not yet implemented. The dual-pathway interference mechanism is hypothesized but not directly probed via residual norm measurement or attention pattern analysis. Mean-pooling destroys sequential structure, limiting RSCE to contexts where ordering is not the primary inference target.

## References

- Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Li, Zhiyuan Liu, and Jie Tang. 2024. LongBench: A bilingual, multi-task benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the 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, and 12 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*.
- Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. xRAG: Extreme context compression for retrieval-augmented generation with one token. In *Advances in Neural Information Processing Systems*.
- Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.
- Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.
- Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.
- Weizhi Fei, Xueyan Niu, Guoqing Xie, Yingqing Liu, Bo Bai, and Wei Han. 2025. Efficient prompt compression with evaluator heads for long-context transformer inference. In *Advances in Neural Information Processing Systems*.
- Yair Feldman and Yoav Artzi. 2025. Simple context compression: Mean-pooling and multi-ratio training. *arXiv preprint arXiv:2510.20797*.
- Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. In-context autoencoder for context compression in a large language model. In *International Conference on Learning Representations*.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.
- Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. 2025. The unreasonable ineffectiveness of the deeper layers. In *International Conference on Learning Representations*.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.
- Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.
- Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023a. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*.
- Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023b. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.
- 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*.
- Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. 2025. Prompt compression for large language models: A survey. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics*.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*.
- Sheng Liu, Haotian Ye, Lei Xing, and James Y. Zou. 2024b. In-context vectors: Making in context learning more effective and controllable through latent space steering. In *International Conference on Machine Learning*.
- Tianyang Liu, Canwen Xu, and Julian McAuley. 2024c. RepoBench: Benchmarking repository-level code auto-completion systems. In *International Conference on Learning Representations*.

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*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Jesse Mu, Xiang Lisa Li, and Noah D. Goodman. 2023. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems*.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning*.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models’ sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *International Conference on Learning Representations*.

Adam S. Shai, Sarah E. Marzen, Lucas Teixeira, Alexander Gietelink Oldenziel, and Paul M. Riechers. 2024. Transformers represent belief state geometry in their residual stream. In *Advances in Neural Information Processing Systems*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. 2024. Function vectors in large language models. In *International Conference on Learning Representations*.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang

Wang, and Beidi Chen. 2023. H<sub>2</sub>O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Hua, Josephine Li, Amanda Askell, Anna Jones, Nat DasSarma, Ethan Perez, Saurabh Ghaisas, and Dan Hendrycks. 2023. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*.

## A Baseline Implementation Details

Both baselines share infrastructure with RSCE: identical sample indices, prompt templates (Context: / Question: / Answer: for QA; # Cross-file context: / # Current file: / # Complete the next line: for code), and generation parameters (greedy, max\_new\_tokens=50, bfloat16, H100-80GB). Failed compression attempts record uncompressed token counts, preserving benchmark integrity.

LongLLMLingua (Jiang et al., 2024) uses `NousResearch/Llama-2-7b-hf` as a separate compressor (EMI); QA uses `condition_compare=True, rank_method=longllmlingua, reorder_context=sort`; code disables question-aware mode. Three progressively relaxed parameter bundles are attempted on split-document and merged-context inputs.

EHPC (Fei et al., 2025) is implemented in NMI mode (same model for compression and inference): a 50-probe NIAH pilot per model selects top-8 evaluator heads; targeted forward hooks capture only the evaluator layer (8 GiB budget); observation-window rows are summed and smoothed with a 1D pooling kernel (size 3); prompts reconstruct from retained token IDs with character-offset accounting to prevent BPE artifacts. EHPC’s RepoBench-C failures (40–50%) stem from the attention memory budget being exceeded on long code sequences; all reported EditSim values are computed over successful samples only.

## B Per-Task QA Breakdown

HotpotQA consistently shows higher RSCE retention than 2WikiMultiHopQA. HotpotQA requires bridging two supporting facts—a structure the residual encoding’s global semantic frame can partially represent. 2WikiMultiHopQA requires longer multi-hop chains where sequential token

Table 6: RSCE per-task QA breakdown for all five models. HotpotQA  $n = 17$ ; 2WikiMultiHopQA  $n = 91$ . \*Vec+F exceeds baseline on HotpotQA due to Qwen’s low baseline (0.226).

Model	Task	Base F1	Q-only	Vec	Vec+F	Ret.
LLaMA-8B	HotpotQA	0.688	0.378	—	0.493	72%
LLaMA-8B	2WikiMQA	0.358	0.269	—	0.303	85%
Mistral-24B	HotpotQA	0.622	0.425	0.455	0.554	89%
Mistral-24B	2WikiMQA	0.534	0.208	0.188	0.316	59%
DeepSeek-14B	HotpotQA	0.495	0.109	0.118	0.328	66%
DeepSeek-14B	2WikiMQA	0.313	0.175	0.174	0.193	62%
LLaMA-70B	HotpotQA	0.627	0.521	0.521	0.583	93%
LLaMA-70B	2WikiMQA	0.600	0.261	0.232	0.325	54%
Qwen-7B	HotpotQA	0.226	0.231	0.223	0.259	115%*
Qwen-7B	2WikiMQA	0.139	0.088	0.108	0.063	45%

precision matters more. The LLaMA-70B HotpotQA result (93% retention) demonstrates that at large scale, RSCE can approach full-context performance on simpler multi-hop tasks.

# Tricking Open-World Object Recognition Models: Uncertainty in Out-of-Distribution Detection

Wout Teillers and Matias Valdenegro-Toro

Bernoulli Institute, Faculty of Science and Engineering, University of Groningen  
w.j.a.teillers@student.rug.nl, m.a.valdenegro.toro@rug.nl

## Abstract

Object recognition models are well studied on benchmark datasets, typically focusing on performance in retrieving objects that exist in images. However, in real-life scenarios there is no prior knowledge of an object’s existence, and current research fails to assess model performance in these situations. This research aims to shed light on this problem by testing three Open-World models, YOLO-World, Grounding Dino and GPT-4o, on the LVIS, Open Images, and JUS datasets. We design an experiment where models are confronted with impossible prompts by instructing them to retrieve non-existing objects. This allows us to observe the models’ uncertainty performance. Overall, GPT-4o performed poorest with regard to object recognition and uncertainty estimation. GPT-4o showed to be highly overconfident. In contrast, YOLO-World and Grounding Dino are slightly underconfident, but they are superior in their uncertainty calibration in comparison to GPT-4o. However, all three models occasionally assign high confident predictions to non-existing objects. Showing that improvement can still be made to the uncertainty estimation of these models when confronted with impossible prompts.

## 1 Introduction

Object recognition is one of the most important fundamental parts of computer vision, and the last two decades the amount of research on this topic has increased tremendously (Zou et al., 2023). Object recognition is an essential task in computer vision that involves classifying and localizing objects within images or video frames. This technology has found widespread application across numerous domains, such as autonomous vehicles, industrial automation, video surveillance, and various other fields.

Object recognition models first classify zero or more objects within an image. In contrast to image

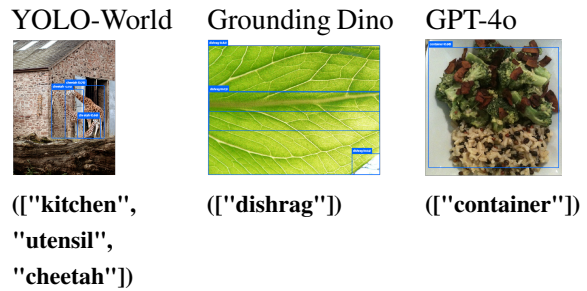


Figure 1: Predictions of missing labels of the three various models, these examples showcase that the models give prediction for non-existing labels with a high confidence. Showing the inability to handle their uncertainty.

classification, these models can recognize multiple objects within a picture, whereas image classification aims to classify a picture as a whole. In addition to classification, object recognition models are capable of the localization of objects through the application of bounding boxes. These bounding boxes show the region where the object can be found within the picture. In Figure 2 it can be seen how a model retrieves objects from an image depending on which classes are prompted. Other examples can be found in Appendix I. Moreover, object recognition models are trained on a dataset with a finite set of object classes. This limits their ability to classify objects that are beyond the classes within the training data. As a result, such objects can not be classified correctly. However, research has found that whenever faced with unknown objects, these models frequently misclassify them as an object class in the training data (Joseph et al., 2021). This behavior is undesirable, as it decreases the reliability of the predictions. Furthermore, research has found that even small modifications to specific sub-regions of an image can affect the model’s ability to detect objects in other, non-local parts of the scene (Rosenfeld et al., 2018). In particular, adding objects whose classes were present during training can lead the model to

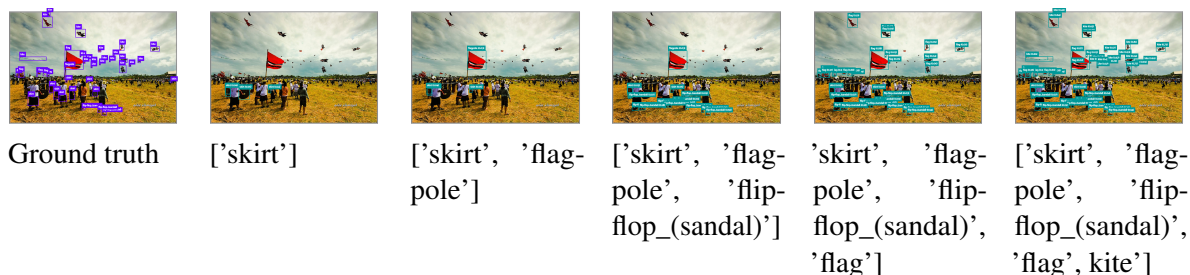


Figure 2: Subset of ground-truth classes evaluated with YOLO-World, illustrating how the model reassigns object labels when it predicts higher confidence for an alternate class.

flip or suppress the classification of other objects in the image. This shows the brittleness of the models to misclassify objects. These misclassifications presents a serious issue, particularly in high-risk scenarios where incorrect detections could lead to severe consequences (Andres et al., 2024).

Current research is focused on the performance of these models on benchmark labeled datasets. The focus lies in detecting objects that are present in the images. However, current research on these models does not take into account what happens when these models try to detect objects that are not present in an image. Object recognition models often fail to correctly refrain from identifying missing object classes or fail to demonstrate its uncertainty correctly as shown in Figure 1. Therefore, this research aims to shed light to this problem, as in real-life scenarios models often don't know if an object is present in the image or not. Therefore, understanding their behavior in these situation is crucial. This gives rise to the following research question and sub research questions:

1. How do Open-World object recognition models perform when confronted with impossible prompts?
2. How do these models react when asked to identify objects that are not visible?
3. To what extent are these models capable of recognizing when an object is absent and cannot be detected?

The goal is that models can correctly show how confident they are with their predictions, also known as their uncertainty. This uncertainty is crucial such that users can trust the output of the models. As such, understanding and addressing the uncertainty in these models is vital for their responsible development. These models function as foundational models as they can be implemented

in a wide range of applications. The goal of this research is to test the knowledge of these models in their open-world capabilities. This is done by exploring the performance and limitation of them in scenarios where models should be highly uncertain by including objects that are absent from the images. The contributions of this work are: an evaluation of several open world object detection models under labels that are prompted but missing from the image, results evaluating calibration and task performance, and our evaluation reveals shortcomings of open world object detection models in terms of calibration under a form of distribution shift.

### 1.1 State-of-the-art

Classical object recognition models are only capable to detect object classes which were present in the training dataset, leading to the inability of detecting unseen objects. However, it is crucial that these models are able to perform out-of-distribution detection (Yang et al., 2024). Several solutions are proposed to address this issue. First of all, open-set object recognition models can recognize unknown objects as "unknown", thereby eliminating the issue of the incorrect labeling of object classes beyond the training data. This addresses the issue of the inability to generalize beyond the scope of trained classes (Joseph et al., 2021). To extend this solution the term 'Open-World Object Recognition models' has been coined. Research on Open-World object recognition aims to solve the issues regarding the limited classes of the training data by allowing the model to learn object classes beyond these classes (Li et al., 2024). For example, Open-World object recognition models can classify objects that are not included in the training data as "unknown". Subsequently, these "unknown" classes can be externally classified, which allows the models to learn these new classes without the need for retraining. Hereby,

open-set and Open-World object recognition models help mitigate the problem of encountering unknown objects.

Furthermore, zero-shot models have been introduced to allow models to identify object classes beyond the classes that it was trained on (Cao et al., 2025). This is done by looking at the proximity of words in the word embedding space. Zero-shot models utilize this principle to allow for the classification of classes outside of the training data. This allows zero-shot models to classify objects it has not seen in training.

The issue of handling unknown objects has been addressed, but uncertainty remains a potential challenge in object recognition. Even though confidence scores are already present, these do not always reflect correct uncertainty. Models must not be overconfident as this can be problematic, especially in high-stakes decision-making scenarios (Valdenegro-Toro, 2021). Therefore evaluating this uncertainty remains crucial.

Utilizing this confidence score increases user trust, however this does not help mitigate model hallucinations. Model hallucinations are high confidence incorrect outputs and reducing them often has a negative impact on object recall (Ren et al., 2024). These hallucinations limit the ability for proper application, as observed in Vision-Language Models (Liu et al., 2024a). Therefore, hallucinations are also major limitation in object recognition.

In conclusion, object recognition models have been developed to address the challenges of out-of-distribution detection. Models incorporate uncertainty scores reflecting the confidence in their predictions. Both features are designed to enhance the trustworthiness of object recognition systems and improve their ability to manage uncertainty.

Since limited research has been done on the behavior of models when faced with impossible prompts, this study aims to address this gap. We will compare open-world object recognition models on benchmark datasets through an experiment in which the models are presented with impossible prompts. Previous work has shown that large language models are prone to hallucinate (Huang et al., 2025) and this problem has also been observed in large vision language models (Sahoo et al., 2024). In general, hallucinations are a significant issue in large foundation models (Jin et al., 2025), therefore, we hypothesize that these foundation object detection models will also hallucinate in these impossible scenarios.

## 2 Methodology

### 2.1 Models

In this research, three models are evaluated on three datasets to aid in understanding the overconfidence level of Open-World object recognition models. Each model has shown individual excellence in the performance of object recognition tasks, by implementing new model designs to overcome the limitations of previous models, and by scoring high mAP scores on benchmark datasets.

The first model that will be used in this experiment is YOLO-World specifically the YOLOv8l-world version. YOLO-World is an Open-Vocabulary Open-World object recognition model that extends the limitation of models trained on a relatively small number of class datasets and extends previous YOLO (You Only Look Once) detectors. This enables this model to recognize objects outside of its initial training data (Cheng et al., 2024). Open-Vocabulary models, such as YOLO-World, are designed to classify objects beyond the predefined classes they were trained on, offering a significant advantage in real-life environments. The weights of the YOLO-World model are imported via the Ultralytics library.

The second Open-World model is Grounding DINO, a model built upon DINO (DETR with Improved deNoising anchOr boxes)(Zhang et al., 2022). Grounding DINO is an Open-Set object recognition model that is capable of recognizing objects given text commands as input (Liu et al., 2024b). The design of Grounding DINO is made to better combine the cross modality information. Hereby, Grounding DINO is able to fuse the text features with the image features, which results in better overall performance such as a 52.5 AP on the COCO (Common Object in Context) detection zero-shot transfer benchmark (Liu et al., 2024b). The Grounding DINO model is implemented using a Hugging Face environment, and the grounding-dino-base model is used in this research.

The final model is GPT-4o, a Generative Pre-trained Transformer, which is a multimodal system capable of processing various types of data including images and text, making it suitable for object recognition tasks (Yang et al., 2023). Trained on a diverse number of data sources, GPT-4o's ability to handle object detection presents an interesting topic for comparison in this study. GPT-4o is accessed using a chat completion API provided by OpenAI, including the image and the prompt. The prompt

(in Appendix C) is designed such that the output format is in line with YOLO-World and Grounding Dino.

This comparison aims to shed light on the capabilities and limitations of these models, particularly in handling impossible prompts and uncertainty.

## 2.2 Datasets

To test these models, this research will make use of the LVIS validation set. The LVIS dataset serves as a widely used benchmark for object recognition tasks and is used to evaluate both the YOLO-World model and the Grounding DINO model. This dataset consists of 164,000 images, spanning 1000 distinct object categories (Gupta et al., 2019).

The second dataset is the Open Images dataset, which is a benchmark dataset used for the training and validation of a wide range of state-of-the-art computer vision tasks (Kuznetsova et al., 2020). The dataset consists of around 9 million images including 600 object categories. This research will use its validation set for its experiment.

Lastly, a dataset containing Japanese uncertainty scenes will be used. This dataset contains pictures designed for testing the uncertainty of Vision-Language Models (VLMs) (Groot and Valdenegro-Toro, 2024), and is therefore expected to give interesting insights into the performance of the state-of-the-art object recognition models. The dataset is available via <https://github.com/ML-RUG/jus-dataset> and contains images, including prompts used for Vision Language Evaluation.

## 2.3 Data Gathering

Both the LVIS and Open Images datasets are imported using a FiftyOne library, which provides an easy interaction with the datasets. In this study, 1000 images of each of these two datasets are tested by setting the *max\_samples* parameter to 1000.

The full dataset of the Japanese Uncertainty Scenes is used, containing 39 images. This dataset does not contain ground truth values for the objects. To manually label these objects, the information in the prompt is used. When no object classes are present in the prompt, the objects in the image will be manually assigned. This ensures that all images have labeled existing objects. The labels assigned for all images can be found in Appendix H.

Each model is tested on all three datasets. For each sample in the dataset, a list of ground truth object classes is retrieved. Afterwards, a list of

missing object classes is generated. The aim for the missing classes is to test the out-of-distribution detection capabilities of the models. This list is generated as follows:

Let  $\mathcal{C}_{LVIS}$  denote the set of classes in the LVIS dataset,  $\mathcal{C}_{OI}$  denote the set of classes in the Open Images (OI) dataset and  $\mathcal{C}_{JUS}$  denote the set of classes in the Japanese Uncertainty Scenes (JUS) dataset.

We then define the missing classes set used for testing as follows:

### 1. Missing classes for testing on Open Images:

$$\mathcal{C}_{miss, OI} = \mathcal{C}_{LVIS} \setminus \mathcal{C}_{OI} \quad (1)$$

### 2. Missing classes for testing on LVIS:

$$\mathcal{C}_{miss, LVIS} = \mathcal{C}_{OI} \setminus \mathcal{C}_{LVIS} \quad (2)$$

### 3. Missing classes for testing on JUS:

$$\mathcal{C}_{miss, JUS} = \mathcal{C}_{LVIS} \setminus \mathcal{C}_{JUS} \quad (3)$$

To further ensure the quality of the missing classes list, we remove common classes such as 'human face' from the  $\mathcal{C}_{LVIS}$ ,  $\mathcal{C}_{OI}$  and  $\mathcal{C}_{JUS}$  as defined in Table 6. These classes might not be labeled but are expected to be present in the images and are therefore removed. The full list of removed classes can be found in Appendix G.

Table 1: Class sets and their description.

Class set	Description
$\mathcal{C}_{LVIS}$	LVIS classes
$\mathcal{C}_{OI}$	Open Images (OI) classes
$\mathcal{C}_{JUS}$	Japanese Uncertainty Scenes (JUS) classes
$\mathcal{C}_{miss, OI}$	Missing classes for OI evaluation
$\mathcal{C}_{miss, LVIS}$	Missing classes for LVIS evaluation
$\mathcal{C}_{miss, JUS}$	Missing classes for JUS evaluation

After computing these unique sets, we further filter the synset lemmas for each object class using Wordnet. Object classes that have a synset relation with the original class are removed. Synsets represent related concepts, so retaining these object classes could reduce the distinctiveness of the missing classes. This process ensures the distinctiveness of the object classes. In our experimental setup, as found in Table 1:

- The set  $\mathcal{C}_{miss, OI}$  is used for retrieving the missing classes for testing on the Open Images dataset.

Table 2: Metrics of Various Models on Different Datasets.

Model	LVIS			OI		
	mAP	AUC	ECE	mAP	AUC	ECE
YOLO-World	0.294	0.85	0.136	0.394	0.82	0.090
Grounding DINO	0.056	0.73	0.032	0.283	0.72	0.024
GPT-4o	0.009	0.83	0.484	0.052	0.87	0.435

- The set  $\mathcal{C}_{\text{miss, LVIS}}$  is used for retrieving the missing classes for testing on the LVIS dataset.
- The set  $\mathcal{C}_{\text{miss, JUS}}$  is used for retrieving the missing classes for testing on the JUS dataset.

This approach to deriving and filtering the set of missing labels is intended to provide a label set suitable for sampling labels that are likely to be absent from the images used during evaluation.

For each sample, both the known and missing classes are evaluated using each model. To maintain balance, the number of missing classes selected matches the number of known classes. Specifically, if the number of known classes is  $n$ , then  $n$  random samples are drawn from the missing class list. To ensure reproducibility, a seed is used for the random number generator.

Lastly, more settings on the project regarding hyperparameters and postprocessing can be found in Appendix J and K.

### 3 Results

**mAP:** In Table 2, the mAP scores of the different models and datasets can be found. The mAP score is calculated using the COCO evaluation protocol (Lin et al., 2014), using an IOU value of 0.5. The mAP scores show the performance of the classification and localization of a model and are used as a standard metric in the evaluation and comparison of object recognition models.

**ROC curves:** In Figure 3, the ROC curves of the models’ predictions are shown. The ROC curve plots the false positive rate against the true positive rate and includes the Area Under the Curve (AUC). It is a useful tool for visualizing the trade-off between precision and recall of a model (Fawcett, 2006). The diagonal line shows random prediction behavior of a model with an AUC score of 0.5. An AUC score of 1.0 indicates perfect discrimination between true positives and false positives. Figure 3 shows the differences between the LVIS and the

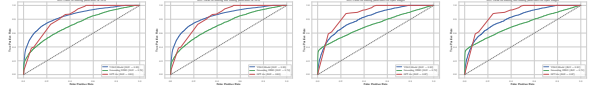


Figure 3: ROC curves for the three models on the LVIS and Open Images datasets. For each dataset, there is one plot displaying the curve for the predictions of the existing labels. The first two plots depict ROC curves of the models tested on the LVIS dataset: the first for existing labels and the second for both existing and missing labels. The last two plots extend this analysis to the Open Images dataset. The plots show the models’ ability to discriminate between true and false positive predictions based on their confidence scores. Grounding Dino shows the poorest performance, with its ROC line closest to the diagonal and the lowest AUC score. Note that there is a marginal difference in the third to fourth significant figure of the AUC score when including the missing labels.

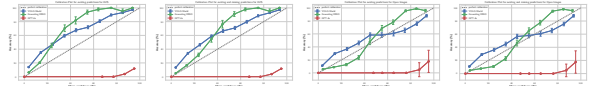


Figure 4: Calibration plots for three different models evaluated on the LVIS and Open Images datasets. These plots visualize the correspondence between predicted confidence scores and actual accuracy. The first two plots depict model calibration using the LVIS dataset: the first for existing labels and the second for both existing and missing labels. The last two plots extend this analysis to the Open Images dataset. The plots show that GPT-4o is highly overconfident and both YOLO-World and Grounding Dino are slightly underconfident.

Open Images dataset. The figure clearly shows that the choice of dataset can result in a difference in AUC score. Overall, Grounding Dino has the lowest AUC scores for both datasets, showing the worst performance in distinguishing between positive and negative predictions. This results in a higher false positive rate and consequently a lower AUC score. Furthermore, YOLO World shows the best performance on the LVIS dataset and GPT-4o on the Open Images dataset. In Figure 3, both plots of the ROC curve with and without the missing classes predictions are shown. The figure shows no difference in ROC curves when plotted using only the existing predictions and plotted using a combination of the existing and missing predictions.

**Calibration Error:** Figure 4 shows the calibration plot of testing the various models on the three datasets. The grey dotted line shows the perfect calibration where the confidence and the accuracy scores are equal. The error bars are included and

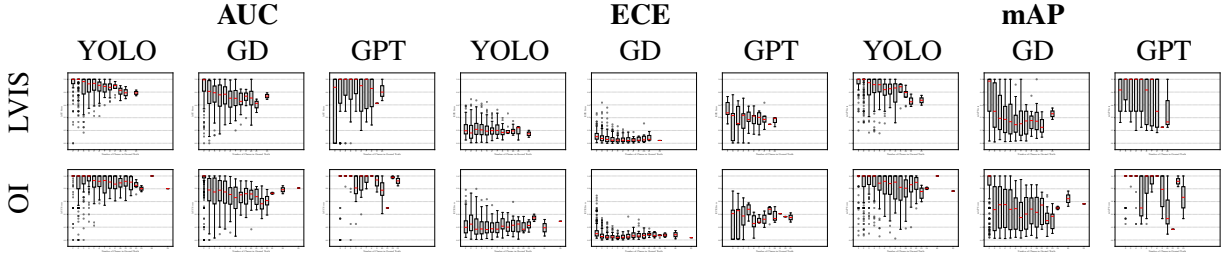


Figure 5: Boxplots of metrics versus the number of ground-truth class labels. The columns show YOLO-World (YOLO), Grounding DINO (GD), and GPT-4o (GPT), and the rows show the LVIS and Open Images (OI) datasets. AUC and ECE variability decreases with more classes, with AUC shifting higher and ECE lower, whereas mAP shows no clear relationship. GPT-4o shows more random behavior.

calculated using following formula:  $\sigma = \sqrt{\frac{p(1-p)}{n}}$ , where  $p$  is the probability representing the confidence scores,  $n$  is the total number of samples, and  $\sigma$  represents the standard deviation. Lastly, the plot is divided into 10 confidence bins, as the default number of bins for the calibration plot. This resulted in well-formed curves and was therefore kept unchanged.

The calibration plot reveals whether a model exhibits over/underconfidence. Models that are positioned mostly below the perfect calibration line are considered overconfident, whilst those above the line are viewed as underconfident. A perfectly calibrated model should align closely with the perfect calibration line. In Figure 4, it can be seen that GPT-4o is entirely under the perfect calibration line and therefore is seen as overconfident for both datasets. In contrast, YOLO-World and Grounding DINO are slightly above the perfect calibrated line for the LVIS dataset and are therefore seen as underconfident. YOLO-World and Grounding Dino both showcase better calibration for the Open Images dataset. Lastly, the results show that over/underconfidence occurs mostly on the LVIS dataset.

Additionally, when plotting the calibration for both existing and missing labels, the models’ calibration is evaluated in a more realistic test scenario. These models are intended for real-world applications where pre-labeled images are not available, meaning it is uncertain whether an object is present in the image. Therefore, comparing predictions with both existing and missing labels provides valuable insights. Figure 4 illustrates these insights, showing that when missing predictions are included, the calibration line of Grounding DINO on the LVIS dataset shifts slightly closer to the perfect calibration line. However, no significant

differences for the other calibration lines are observed.

The calibration errors of the models can be seen in Table 2. The calibration error figures allow for a good visual comparison between the models, but it is also beneficial to have a numeric representation of the calibration error (Guo et al., 2017). Therefore, the Expected Calibration Error (ECE) is used to calculate the numeric calibration errors of the models. The ECE is calculated as follows:

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|, \quad (4)$$

where  $\text{acc}(B_m)$  represents the fraction of true positive samples and  $\text{conf}(B_m)$  represents the average confidence in each bin number  $m$ . In this formula,  $n$  represents the total number of samples and  $m = 10$ , as the calibration errors are divided into 10 bins.

**Distribution of metrics for different class sizes:** In Figure 5, the distribution of the different metrics for different numbers of classes in the ground truth can be seen. The figure shows that the variance of both the AUC score and ECE score decreases when the number of classes increases for YOLO-World and Grounding Dino. The AUC increases and the ECE decreases when the number of classes increases. However, GPT-4o shows no relation between the number of classes and the metrics. Furthermore, the figure shows that there is no relation between the number of classes and the mAP score.

**Confidence Score Distribution:** The distribution of confidence scores of each model on each dataset can be seen in Figure 7. Confidence scores are distinguished based on whether the labels were missing or existing. In Figure 7 it can be seen that GPT-4o predictions are skewed to the left in

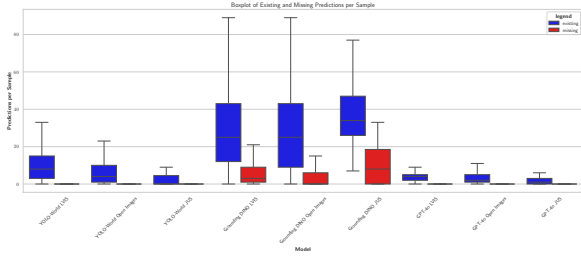


Figure 6: Boxplot illustrating per-sample prediction counts from various models applied to the same dataset. The figure compares the number of existing and missing predictions across models. Overall, Grounding DINO gives the highest number of predictions and all models give fewer predictions for the missing labels than for the existing labels.

terms of their confidence on both the existing labels and missing labels. This demonstrates that GPT-4o frequently provides high-confidence predictions. In contrast, both YOLO-World and Grounding DINO show a right-skewed distribution, showing that most predictions are made with low confidence. YOLO-World, in contrast to Grounding DINO, shows more predictions in the higher confidence bins. These high confidence predictions occur mostly for existing labels, but there are also more high confidence predictions of the missing labels compared with Grounding DINO. The confidence distribution of the missing labels and the existing labels overlap, for greater clarity the separate plots of the distributions can be seen in Figure 9 and Figure 10 in Appendix B.

**Qualitative Comparison:** For a qualitative comparison of the models, a plot of the predictions is shown in Figure 8. This sample received the highest average confidence scores for its existing labels. It can be seen that all three models are able to correctly identify the giraffe in the picture with high confidence. However, Grounding DINO also misidentifies a tree as a giraffe with low confidence. When looking at the prediction for the missing labels it can be seen that both YOLO-World and GPT-4o are able to correctly "see" that the image does not contain these missing labels. However, Grounding DINO does classify a number of objects in the image even though they are not present. These predictions of Grounding DINO do showcase that the model is uncertain about its predictions due to the low confidence scores. More examples can be found in Appendix E.

**GPT-4o Response Error:** The prompt used for receiving the prediction of GPT-4o can be found

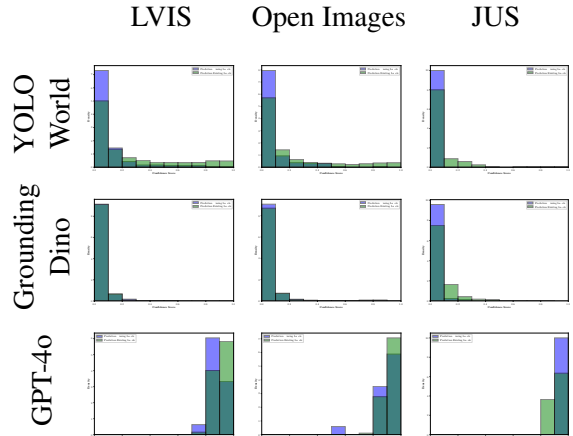


Figure 7: Density histogram showing the distribution of confidence scores for the models' predictions for the missing and existing labels across all datasets. The plots show that GPT-4o exclusively gives predictions at high confidence levels, whereas YOLO-World and Grounding Dino give a wider range of confidence scores, though still showing right-skewed behavior.

Table 3: GPT-4o prompt quality and output consistency. The success rate indicates the percentage of API calls that returned a valid response.

Datasets	Success rate
LVIS	89.2%
Open Images	81.6%
Japanese Uncertainty Scenes	76.9%

in Appendix C. While this prompt resulted in responses in the desired format, GPT-4o remains a Large Language Model (LLM) which occasionally results in hallucinations. Therefore, some images were not processed correctly as the model deviated from the instructions. In Table 3, the accuracy scores can be found for each dataset. For example, GPT-4o provided predictions for 89.2% of the 2,000 samples in the LVIS dataset, with half of the predictions corresponding to existing labels and the other half to missing labels.

## 4 Discussion

A comparison of the three models reveals that GPT-4o exhibits the poorest performance in terms of uncertainty. GPT-4o is notably overconfident and demonstrates poor calibration in representing its uncertainty. Despite achieving the highest AUC score, GPT-4o records the lowest mAP score. This is due to GPT-4o's tendency to generate predictions with high confidence in combination with its inability to correctly localize objects. This contributes

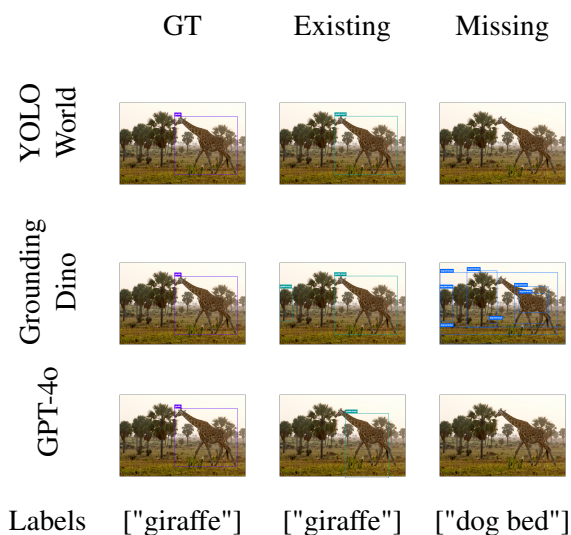


Figure 8: Visual comparison of ground truth (GT), detections of existing labels (Existing), and detections of missing labels (Missing) across different models with the highest average confidence for the existing labels of the LVIS dataset. All three models successfully detect the giraffe in the image, however, Grounding DINO also mislabels a tree. Above all, Grounding DINO is the only model that makes predictions for the missing classes.

to the overconfidence and poor mAP score. Additionally, GPT-4o frequently hallucinates due to defiance with the API call instructions.

Analysis of YOLO-World and Grounding DINO reveals that both models exhibit slight underconfidence for the LVIS dataset. However, both models are calibrated relatively well for the Open Images dataset. In terms of mAP and AUC scores, YOLO-World outperforms Grounding DINO, as it achieves higher values in both metrics. This shows that YOLO-World is superior at recognition and localization of objects and at discriminating between classes. Furthermore, Grounding DINO is more likely to hallucinate missing classes, whilst YOLO-World more often refrains from predicting these classes as seen in Figure 6.

Lastly, even though there are some manually removed incorrect missing labels in the experiment setup, as mentioned in Appendix K. It does not influence the results, due to the large difference between performance metrics in Table 2.

## 5 Conclusions and Future Work

To answer RQ1, the performance of these models when faced with impossible prompts varies across models. GPT-4o rarely predicts missing labels, but when it does, it assigns them high confidence.

This is undesirable, as it fails to adequately convey uncertainty, particularly in response to impossible prompts. YOLO-World and Grounding DINO show a difference in their confidence scores when faced with existing or missing labels, showcasing the ability to give uncertainty estimation. This is shown by relatively good calibration of these models, where both models are only slightly underconfident. To address RQ2, GPT-4o struggles with object localization. Grounding DINO and YOLO-World both demonstrate an ability to represent their uncertainty. However, Grounding DINO’s performance is hindered by a high number of unwanted predictions. In contrast, YOLO-World achieves the best performance in object detection and the most accurate representation of uncertainty when faced with impossible prompts. To answer RQ3, we can see that both YOLO-World and Grounding DINO show good discriminability between existing and missing classes. Where both models are only slightly underconfident. In contrast, GPT-4o shows poor performance as the model is highly overconfident. GPT-4o does show a high AUC score, this is likely due to the fact that the model produces a low number of predictions. Therefore, this AUC score does not represent the performance of GPT-4o accurately. In conclusion, YOLO-World and Grounding DINO show an ability to express uncertainty, while GPT-4o fails to do so correctly. However, improvements to the uncertainty estimations can still be made. Enhancing their performance in this regard is crucial for their application in real-life high-stakes scenarios.

**Limitations.** An important next step is to evaluate the models on a larger dataset such as the full validation set of LVIS and Open Images, this will better assess their performance on a wider range of scenarios. We observed that some missing classes were actually present in the image but not labeled, this resulted in unwanted predictions, these were removed from the results. For future implementation robustness can be enhanced by preventing these errors. In this research there is no contrast made whether missing classes are present in the training data of a model, however, this could potentially bias results. Lastly, a limitation with the ROC curves is that these are constructed using only the true positive and false positive predictions. In object detection the true negatives and false negatives are ill defined and are consequentially not used in the ROC curves. Therefore, the AUC values should be interpreted with caution.

## References

- Alain Andres, Aitor Martinez-Seras, Ibai Laña, and Javier Del Ser. 2024. On the black-box explainability of object detection models for safe and trustworthy industrial applications. *Results in Engineering*, 24:103498.
- Weipeng Cao, Xuyang Yao, Zhiwu Xu, Ye Liu, Yinghui Pan, and Zhong Ming. 2025. A survey of zero-shot object detection. *Big Data Mining and Analytics*, 8(3):726–750.
- Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. 2024. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16901–16911.
- Tom Fawcett. 2006. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.
- Tobias Groot and Matias Valdenegro-Toro. 2024. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models. *arXiv preprint arXiv:2405.02917*.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.
- Agrim Gupta, Piotr Dollar, and Ross Girshick. 2019. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.
- Haibo Jin, Peiyan Zhang, Peiran Wang, Man Luo, and Haohan Wang. 2025. From hallucinations to jailbreaks: Rethinking the vulnerability of large foundation models. *arXiv preprint arXiv:2505.24232*.
- KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. 2021. Towards open world object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5830–5840.
- Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Mallocci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. 2020. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*.
- Yiming Li, Yi Wang, Wenqian Wang, Dan Lin, Bingbing Li, and Kim-Hui Yap. 2024. Open world object detection: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(2):988–1008.
- Tsung-Yi Lin, Licheng Ma, and Serge Belongie. 2014. *Coco dataset: Detection evaluation*.
- Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024a. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*.
- Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, and 1 others. 2024b. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer.
- Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, and 1 others. 2024. Grounding dino 1.5: Advance the "edge" of open-set object detection. *arXiv preprint arXiv:2405.10300*.
- Amir Rosenfeld, Richard Zemel, and John K. Tsotsos. 2018. *The elephant in the room*. *Preprint*, arXiv:1808.03305.
- Pranab Sahoo, Prabhash Meharia, Akash Ghosh, Sriparna Saha, Vinija Jain, and Aman Chadha. 2024. A comprehensive survey of hallucination in large language, image, video and audio foundation models. *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11709–11724.
- Matias Valdenegro-Toro. 2021. I find your lack of uncertainty in computer vision disturbing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1263–1272.
- 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*. *Preprint*, arXiv:2310.11441.
- Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. 2024. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12):5635–5662.
- Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. 2022. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*.
- Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. 2023. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276.

## A Broader Impact Statement

Uncertainty estimation for large models is socially relevant as it is desirable to detect hallucinations or incorrect predictions, which is needed for critical applications. This paper reveals that even open world object detection models struggle with uncertainty estimation when object classes are prompted but not present and are miscalibrated and often overconfident.

Uncertainty estimation and Computer vision models require extensive experimental validation with data representative of the use case before being used in real-world applications, and unfortunately there are no guarantees on uncertainty estimation quality and performance.

## B Separate Density Histogram Confidence Scores

Figures 9 and 10 show the density plots of the confidence scores for the existing and missing confidence scores are shown in separate figures to see the differences more clearly.

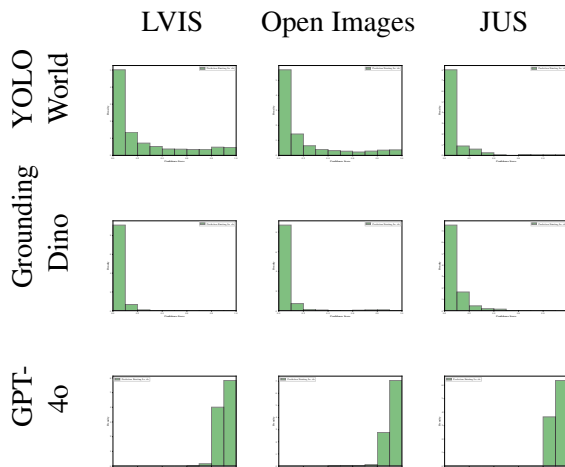


Figure 9: Density histogram showing the distribution of confidence scores for the model predictions of the existing labels across all datasets. The plots show that GPT-4o only gives high confidence predictions, whereas YOLO-World gives a broader range of scores whilst still being right skewed. Furthermore, Grounding Dino, whilst also predicting in a larger range, gives lower confidence predictions more often.

## C Prompt GPT-4o

In Figure 11 the prompt used for the GPT-4o API. The prompt is designed to let GPT-4o give predictions similar to YOLO-World and Grounding Dino.

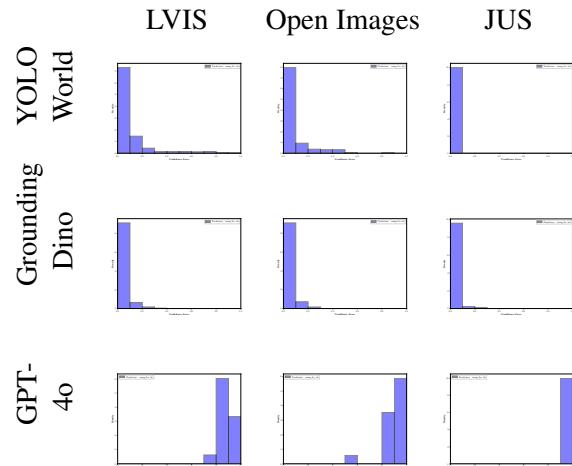


Figure 10: Density histogram showing the distribution of confidence scores for the model predictions of the missing labels across all datasets. The plots show that GPT-4o only gives high confidence predictions for the missing labels, whilst YOLO-World and Grounding Dino give predictions in a wider range but mainly giving low confidence predictions.

The json output of the response API is parsed and saved for further processing. More details about the implementation can be found in the code repository found in Appendix D.

## D Code Repository

<https://github.com/WoutTeillers/openworld-uncertainty-ood.git>

In the GitHub repository (in the url above) the code used in the experiment can be found which include the scripts for running the models and the notebooks for visualizing the results. The scripts are executed on "python3.10.11" and the necessary libraries and their versions can be found in the "requirements.txt" file. Lastly, an API is used for the predictions of GPT-4o which requires a valid OpenAI API key.

## E Qualitative comparison

This section provides examples of the prediction of the various models on the different datasets. These examples give visual insights in their performance.

Figure 16 shows the highest average confidence scores for the missing labels of the Open Images dataset. It shows that GPT-4o gives a high prediction for a missing label in the image, whereas the other two models correctly refrain from classifying the missing labels. Furthermore the Figure shows

You are an object recognition model capable of detecting and localizing objects within an image. Given an image with width = {img\_width} and height = {img\_height}, you will receive a list of object classes that I want you to detect.

Your task is to find all objects in the image that match these class labels and provide the following details for each object:

1. The confidence score (ranging from 0 to 1) of the detection, ensuring that only objects with a confidence score greater than or equal to {conf\_threshold} are included.
2. The bounding box for each detected object, given by the center ( $x\_center, y\_center$ ) of the bounding box (in pixel coordinates) and the width and height of the bounding box (in pixel width). The bounding box should tightly enclose the object and should be calculated with respect to the object's aspect ratio and position.
3. The class label for each object, corresponding to one of the classes in the provided list.

Make sure that each object is localized as accurately as possible within the image. The origin point (0, 0) is at the top-left corner of the image.

The format for your response should be a JSON string like the following:

```
{'scores': [], 'boxes': [[x_center, y_center, width, height]], 'labels': [String]}
```

Where:

- scores: A list of confidence scores for each detection.
- boxes: A list of bounding boxes for each object, where each bounding box is a list of four values:  $[x\_center, y\_center, width, height]$ , representing the center of the bounding box and its dimensions.
- labels: A list of strings, where each string is the class label for the corresponding object in the image.

Please ensure that you only include detections that meet the confidence threshold and that the bounding boxes are as precise as possible, accurately matching the position of each object in the image. And that the output contains only the JSON without comments.

Figure 11: Prompt used for the GPT-4o API. In this prompt, the variables `img_width`, `img_height`, and `conf_threshold` are dynamically assigned based on the image dimensions and the specified confidence threshold.

the difference in the models' prediction capabilities. Figure 16 shows that GPT-4o provides only a few predictions, but with inaccurate localization. In contrast, Grounding DINO generates a larger number of predictions with varying confidence levels. YOLO-World makes correct predictions for several objects but, interestingly, fails to predict relatively clear classes, such as the wheels of the truck.

Figure 12 provides a clearer view of GPT-4o's difficulty in localizing objects within an image. This issue contributes to GPT-4o's low mAP score as seen in Table 2, as most of its predictions are false positives due to the poor localization.

Figures 13 and 14 illustrate instances where YOLO-World and Grounding DINO assign high confidence scores to missing labels. These examples reveal that the models tend to assign high confidence when there are semantic similarities between two classes or when the object class names are similar. In Figure 13, YOLO-World gives a high

confidence prediction for "pickup\_truck" because it shares a strong semantic similarity with 'car'. However, it assigns a lower confidence score to "pickup\_truck", indicating that YOLO-World recognizes it resembles a car more than a pickup truck. Additionally, in Figure 14, the model assigns a high confidence prediction for the missing class "cutting board". This is because both "cutting board" and "surfboard" share the common feature of being "boards". Once again, the model assigns higher confidence to the existing class, demonstrating that the model predicts the object class "surfboard" as more likely.

Figure 15 shows the sample with the most missing predictions on the JUS dataset. Where only Grounding DINO incorrectly predicts the missing class "pillow". The model assigns the entire group of statues the label "pillow" in addition with a number of predictions for the individual statues. This is inline with the prediction of the existing labels.

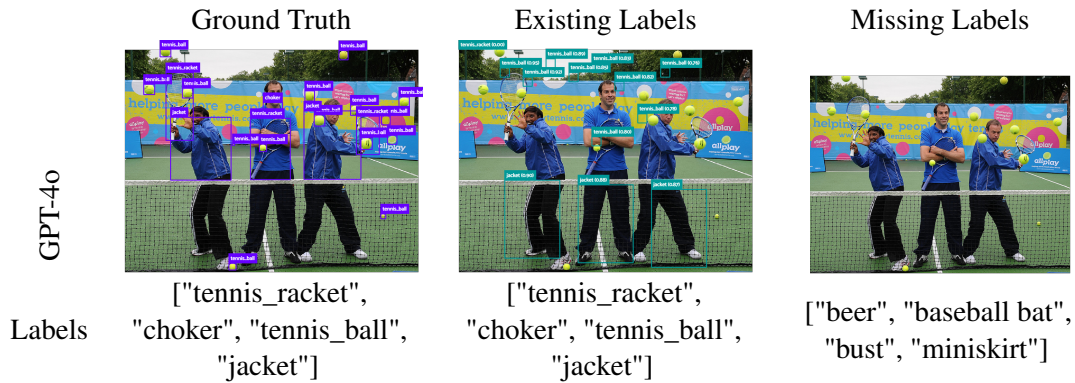


Figure 12: Visual comparison of ground truth, detections, of the image with most false positive predictions of GPT-4o. Showing the inability of GPT-4o to correctly localize objects in an image.

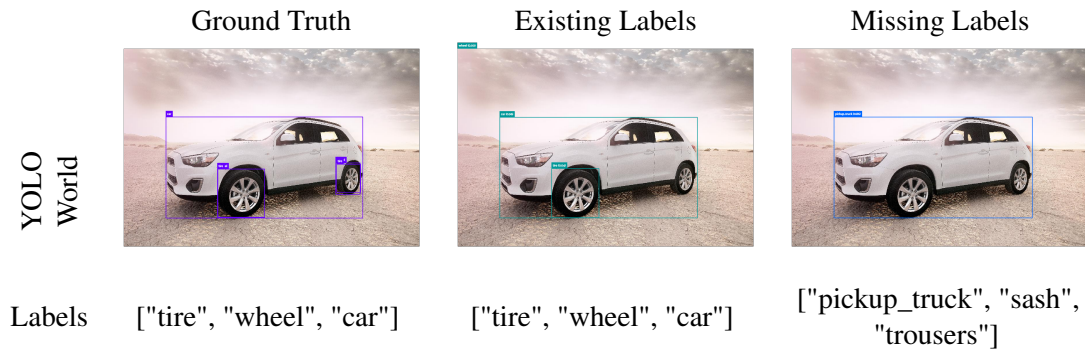


Figure 13: Visual comparison of ground truth, detections, of the image with highest average confidence for the missing predictions of YOLO-World on the Open Images dataset. YOLO-World shows a high confidence prediction for the missing label "pickup\_truck", likely due to semantic similarity between the ground truth label "car". The model does show a difference in confidence for these predictions.

Grounding DINO also gives a prediction to the entire group, in combination with prediction on the individual statues. However, for the existing labels, Grounding DINO shows much higher confidence scores. Furthermore, GPT-4o gives a few high confidence predictions on the existing labels. The predictions are not properly localized and GPT-4o misses a large number of predictions for statues. Also, YOLO-World is not able to recognize any of the statues in the image. YOLO-World and GPT-4o both correctly refrain from predicting the missing label in this image.

The main findings of the qualitative comparisons are that Grounding DINO gives the most predictions of the three models, thereby missing fewer objects in the image but occasionally labeling incorrect objects. The difference between the number of predictions between the models can be seen in Figure 6. Additionally, Grounding DINO shows a clear distinction in its confidence for existing and missing labels. GPT-4o shows poor performance in image localization, with its predictions always

being of high confidence, which is in line with the density distribution shown in Figure 7. Furthermore, this model does not classify all the desired objects in an image consistently. YOLO-World shows good distinguishability between existing and missing labels, but occasionally misses some of the existing ones.

## F Incorrect Missing Class Labels

Table 4 and 5 the incorrect missing labels can be found for the LVIS and Open Images dataset. The predictions for these labels are removed from the specific images to ensure that the predictions are for objects that are actually not present.

Figure 17 shows this problem, the image contains a numerous labeled object classes, but the class of "power plugs and sockets" is not labeled in the image resulting in this complication.

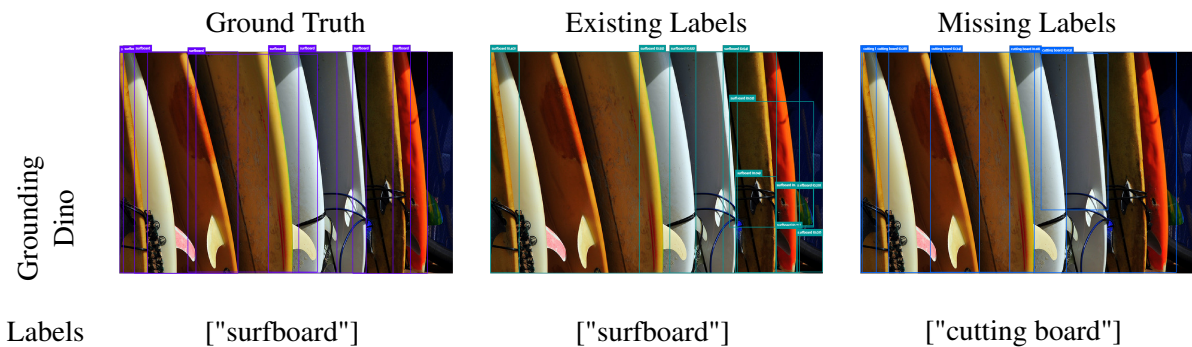


Figure 14: Visual comparison of ground truth, detections, of the image with highest average confidence for the missing predictions of Grounding Dino on the LVIS dataset. Grounding Dino gives high confidence predictions for the missing labels, likely due to the fact that both the ground truth label "surfboard" and the missing label "cutting board" are both type of "boards". The models does show difference in confidence level for these predictions.

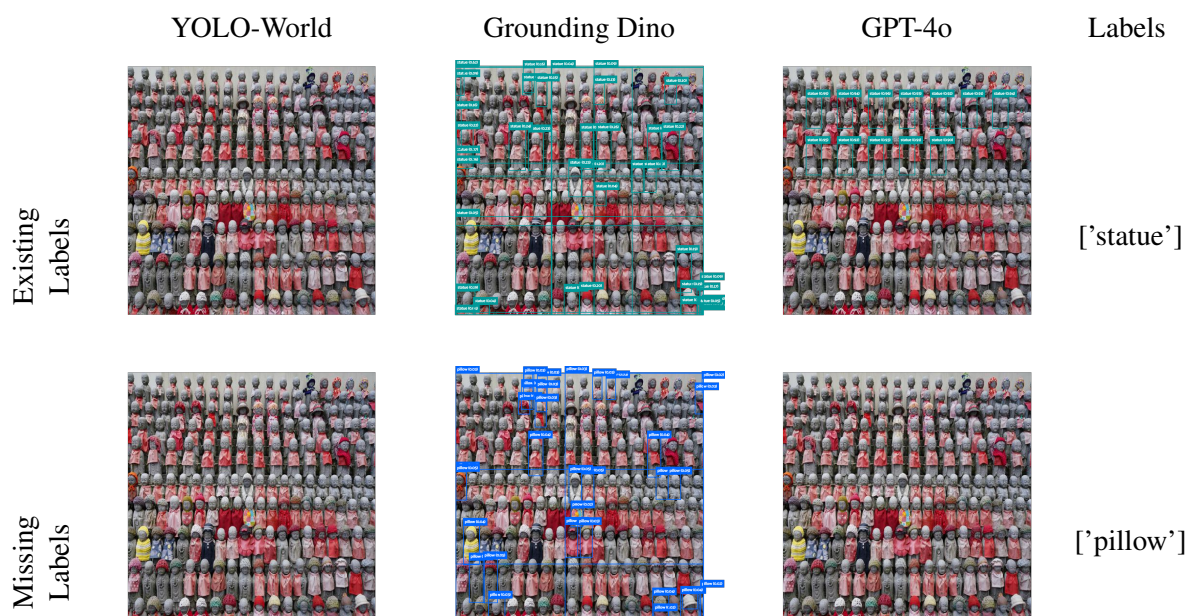


Figure 15: Visual comparison of ground truth, detections, and missing labels across the different models with the highest average confidence for the missing labels of the Japanese Uncertainty Scenes dataset. YOLO-World gives no predictions for both the existing and missing labels, whereas Grounding DINO gives a high number of predictions for both. GPT-4o shows a small number of incorrectly localized predictions for the existing labels and no prediction for the missing labels.

	Ground Truth	Existing Labels	Missing Labels
YOLO World			
Grounding Dino			
GPT-4o			
Labels	["wheel", "land vehicle", "vehicle", "truck", "person", "tire", "car"]	["wheel", "land vehicle", "vehicle", "truck", "person", "tire", "car"]	["hairgrip", "freight_car", "chalice", "headphone", "cooking_pan", "dixie_cup", "sweater"]

Figure 16: Visual comparison of ground truth, detections, and missing labels across the different models with the highest average confidence for the missing labels of the Open Images dataset. The three models show notably differences in their predictions for the existing labels and GPT-4o is the only model that gives a prediction for the missing labels.

## G Common Classes Removed from Class List

A list of common classes were removed from the list of classes as these classes are expected to be found in the images even though they might not be labeled. In Table 6 the list of classes that was removed can be found for the LVIS, Open Images and Japanese Uncertainty Scenes (JUS) datasets. Note that there were no classes removed from the JUS dataset.

## H Class labels for Japanese Uncertainty Scenes

The ground truth values for the Japanese Uncertainty Scenes were manually assigned. In Table 7 the ground truth labels for each image can be found. The JUS dataset can be found at <https://github.com/ML-RUG/jus-dataset>.

## I Prediction per subset of Labels

Figure 18 illustrates model predictions on a subset of the ground truth classes. In this figure predictions are shown where one class is added to the prompt at each step. This shows how the models

make predictions based on the desired labels. It is evident that the models can give incorrect labels with low confidence when the correct label is not available yet. Whenever the label gets prompted it can be seen that the model changes its prediction.

## J Hyperparameters

Running the object recognition models there is one key hyperparameter to tune, the confidence level threshold. This threshold indicates above which confidence level predictions are accepted. Hereby, allowing to manually tune the most optimal threshold to allow the correct predictions to pass and unwanted prediction to be omitted. During this experiment, the lower predictions give insights into the model's confidence level. Therefore, in the experiment this confidence threshold is set to a small value to achieve a wide range of confidence levels. The threshold is set to get almost all predictions to receive data on the performance of the models. This confidence level can not be set to 0, due to the fact that all prediction will be showcased, resulting in too much unwanted predictions. For consistency across models, this hyperparameter is set to 0.02 for all models, ensuring that predictions fall within the

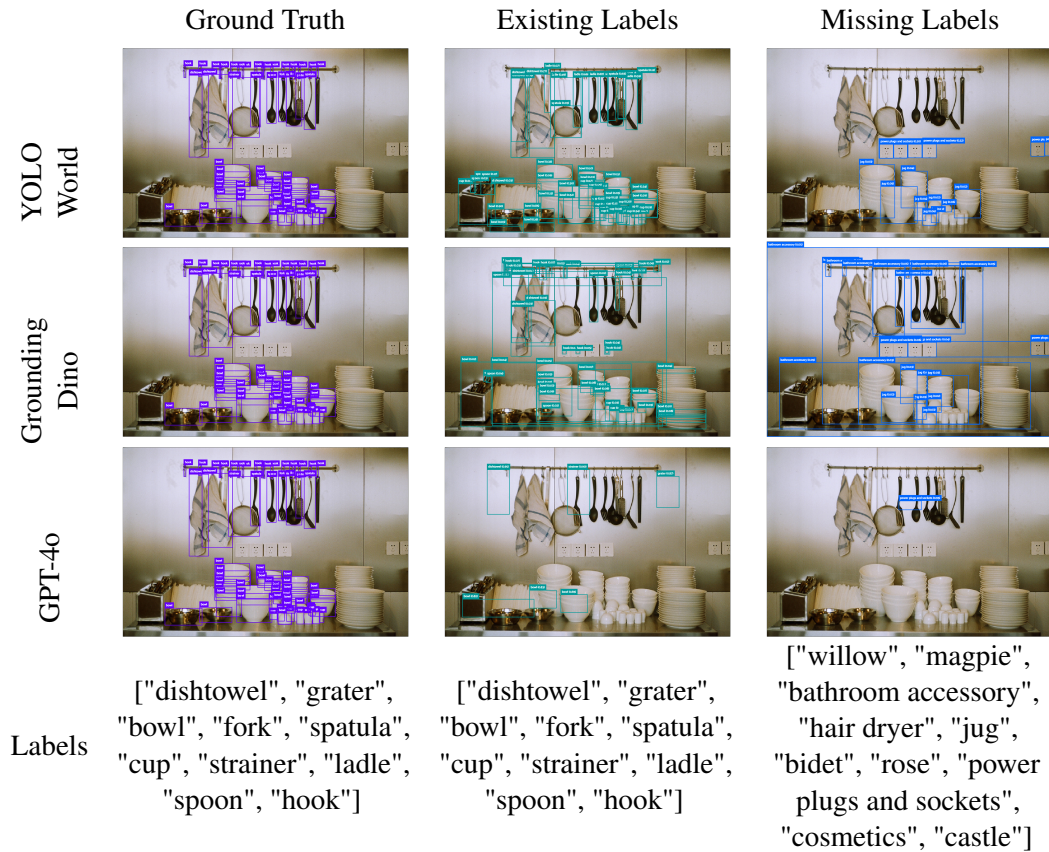


Figure 17: Visual comparison of ground truth, detections, and missing labels across the different models showcasing an incorrect missing label in the image. Showing that classes that are not present in the ground truth can be present in the image. This can lead to incorrect labels in the missing classes list such as "power plugs and sockets" in this example.

same confidence range (0.02-1.00) and minimizing the number of unwanted predictions.

## K Post Processing

Running the Grounding DINO base model on the datasets, resulted in some unwanted predictions. The model occasionally gave prediction where a new combined label was used. For example given the two classes ["dog", "bed"], the model could give a prediction of "dog bed". These predictions were removed as they were not of interest in this study.

Furthermore, the Grounding DINO base model did not make use of Non-maximum Suppression (NMS). This lead to the model giving many overlapping predictions. Due to the fact that YOLO-World did make use of NMS, it was decided to implement the NMS algorithm for the predictions made by Grounding-DINO. This to make sure of a fair comparison between the models. This study used the same IOU value as YOLO-World for a fair comparison. YOLO-World uses an IOU thresh-

old of 0.5, so this value was sub-sequentially used for Grounding DINO. NMS filters predictions of the same label with an IOU value exceeding the specified threshold, retaining the prediction with the higher confidence score.

As GPT-4o did not have many overlapping detections, NMS was not applied to the predictions of this model.

A minor issue with the experiment setup is that while requiring the list of missing classes for each image, it sometimes occurs that the object of that class are present in the image, even-though it was not labeled in the original dataset. This will cause the models to recognize the object with high confidence which reduced the reliability of this experiment, as the experiment assumes that these missing classes are not present in the image. To resolve this, the predictions of the missing labels are manually checked for validity and the missing classes that are present in the image are removed from the predictions. This causes a slight imbalance between the number of existing classes and missing classes



Figure 18: Subset of ground-truth classes evaluated with YOLO-World and Grounding Dino, illustrating how the models reassigns object labels when it predicts higher confidence for an alternate class.

in an image. However, this problem only occurs for 4.3% of all the tested images. The prediction of classes that were removed of specific images can be found in Appendix F, including a visual example that illustrates this issue.

Table 4: Incorrectly retrieved missing labels of the LVIS dataset. The predictions of each label have been removed from the specified image.

Image	Missing LVIS Labels
000000192722.jpg	['footwear']
000000569652.jpg	['power plugs and sockets']
000000452334.jpg	['home appliance', 'kitchen appliance']
000000006777.jpg	['stairs']
000000269417.jpg	['kitchen appliance']
000000394879.jpg	['coffeemaker', 'dessert']
000000375317.jpg	['gas stove', 'home appliance']
000000244965.jpg	['mammal']
000000125247.jpg	['candy']
000000090122.jpg	['kitchen utensil']
000000175205.jpg	['dairy product']
000000290911.jpg	['vehicle']
000000497875.jpg	['tree']
000000424044.jpg	['baked goods', 'mixing bowl']
000000216863.jpg	['picture frame']
000000051618.jpg	['tableware']
000000263589.jpg	['building']
000000044611.jpg	['trousers']
000000048432.jpg	['land vehicle']
000000277858.jpg	['mammal']
000000205055.jpg	['building', 'footwear']
000000406013.jpg	['plastic bag']
000000534751.jpg	['furniture']
000000244157.jpg	['tree']
000000094052.jpg	['microwave oven', 'picture frame', 'food']
000000471842.jpg	['kitchenware']
000000261893.jpg	['bicycle wheel']
000000364210.jpg	['baked goods']
000000442298.jpg	['furniture', 'mammal']
000000468917.jpg	['microwave oven']
000000362140.jpg	['fashion accessory', 'picture frame']
000000555273.jpg	['picture frame']
000000127100.jpg	['footwear']
000000372980.jpg	['mammal']
000000043692.jpg	['countertop', 'furniture']
000000315902.jpg	['furniture']
000000505152.jpg	['kitchen utensil']
000000223032.jpg	['mammal']
000000070164.jpg	['remote control']
000000356153.jpg	['food']
000000334352.jpg	['sports equipment']
000000455691.jpg	['countertop']
000000526794.jpg	['home appliance']
000000086208.jpg	['snack']
000000103223.jpg	['home appliance']
000000501247.jpg	['land vehicle']
000000160142.jpg	['furniture']
000000521200.jpg	['hand dryer']
000000040930.jpg	['window']
000000120527.jpg	['sports equipment']
000000460442.jpg	['fruit', 'footwear']
000000423161.jpg	['flower']
000000053037.jpg	['coffeemaker', 'microwave oven']
000000218751.jpg	['land vehicle']
000000361497.jpg	['aircraft']
000000278303.jpg	['mammal']
000000095841.jpg	['footwear']
000000018090.jpg	['footwear']
000000457737.jpg	['furniture']
000000404698.jpg	['power plugs and sockets']
000000513604.jpg	['tin can']
000000019441.jpg	['tableware']
000000442875.jpg	['furniture']
000000007288.jpg	['bidet']
000000071726.jpg	['soap dispenser', 'cabinetry']

Table 5: Incorrectly retrieved missing labels of the Open Images dataset. The predictions of each label have been removed from the specified image

Image	Missing OI Labels
00141571d986d241.jpg	['hand_towel', 't-shirt']
00146ba1e50ed8d8.jpg	['cylinder']
0035c28612c035fd.jpg	['green_bean']
00acf53b127218c2.jpg	['radiator_grille']
00dc0530e6779ca6.jpg	['baby_buggy']
01491bf840ae9939.jpg	['activewear']
015f5cd905204962.jpg	['trousers']
0197df7725980004.jpg	['rearview_mirror']
01b405e0cab3add3.jpg	['baseball_cap']
01f26ca52e27a8d9.jpg	['pencil_case']
023a57536e17b7b1.jpg	['figurine']
025ffa27eb2ba851.jpg	['printing_machine']
030033e1b4137e3b.jpg	['dog_collar']
03650b9fde97f523.jpg	['wristwatch']
049720d842de2d3e.jpg	['paper_towel']
04d9284ebdc41aeb.jpg	['cordial']
04ec0b057014a648.jpg	['jockey_cap']
006f87bf928f9ba3.jpg	['jewellery']
00c9616a917be867.jpg	['fin_(footwear)']
01c79b8cc239037d.jpg	['wedding_ring']
038ee0bf31929792.jpg	['flip-flop_(sandal)']
05d69a9470032674.jpg	['sport_shirt']

Table 6: List of common classes removed from list of classes.

Class set	List of removed classes
$\mathcal{C}_{LVIS}$	['human arm', 'human beard', 'human body', 'human ear', 'human eye', 'woman', 'man', 'human face', 'human foot', 'human hair', 'human hand', 'human head', 'human leg', 'human mouth', 'human nose']
$\mathcal{C}_{OI}$	['human']
$\mathcal{C}_{JUS}$	[]

Table 7: Ground truth labels of the images in Japanese Uncertainty Scenes dataset.

Filename	List of ground truth class labels
20180728_204527.jpg	['food']
20180729_152751.jpg	['dessert']
20180730_132300.jpg	['sushi']
20180801_203824.jpg	['okonomiyaki', 'food']
20180808_220402.jpg	['sushi']
20180812_145111.jpg	['egg fried rice', 'fried chicken', 'gyoza', 'japanese food']
20180814_181327.jpg	['person', 'food']
DSC01703.jpg	['drawing', 'animal']
DSC01754.jpg	['octopus', 'person']
DSC01796.jpg	['paper lanterns']
DSC01851.jpg	['statue']
DSC01874.jpg	['tombstone']
DSC02396.jpg	['fish']
DSC02711.jpg	['bird', 'building']
DSC02941.jpg	['lantern', 'building', 'paper lanterns']
DSC02960.jpg	['lamp', 'paper lanterns']
DSC03113.jpg	['gate', 'torii']
DSC03256.jpg	['gate', 'torii']
DSC03391.jpg	['bamboo tree']
DSC03397.jpg	['bus', 'bridge']
DSC04631.jpg	['wooden plaque', 'ema']
DSC04742.jpg	['tree', 'building']
DSC04746.jpg	['building']
DSC04796.jpg	['bridge']
DSC04858.jpg	['boat', 'mountain']
DSC05168.jpg	['tree', 'painting']
DSC05403.jpg	['person', 'tree']
DSC05406.jpg	['person', 'tree']
DSC05439.jpg	['warrior']
DSC05535.jpg	['japanese food']
DSC06071.jpg	['tree']
DSC07641.jpg	['coach', 'train']
P9250145.jpg	['tokyo tower']
P9250156.jpg	['building']
P9280467.jpg	['light poles', 'signs']
P9301022.jpg	['building', 'tree']
PA011063.jpg	['volcano']
PA011245.jpg	['statue', 'animal']
PA041459.jpg	['building', 'tree']

# Knowledge Localization and Editability in Small Language Models: A Multi-Stage Experimental Study

Pranamy Nilesch Deshpande

Togo AI Labs

pranamyadeshpande14@gmail.com

Aiswarya Konavoor

Togo AI Labs

aiswarya@togolabs.ai

Sreedath Panat

Togo AI Labs

sreedath@togolabs.ai

## Abstract

The internal mechanisms by which transformer-based language models encode and retrieve factual knowledge remain poorly understood, particularly for small language models (SLMs) operating in the 2–3 billion parameter range. This paper presents a systematic, multi-stage empirical investigation into knowledge localization, compression effects, and knowledge editability across four SLMs—Gemma-2B, Llama-3.2-3B-Instruct, Qwen-2.5-3B-Instruct, and Phi-2—with Meta-Llama-3-8B serving as a large-model baseline. Stage 1 employs causal tracing with activation patching on the CounterFact dataset (~450–500 validated facts per model) to identify the layer or layers most causally responsible for factual recall. Stage 2 compares knowledge density, layer concentration, and redundancy between the 2–3B models and the 8B baseline to quantify the structural effects of model compression on knowledge storage. Stage 3 applies the Rank-One Model Editing (ROME) algorithm at the causally identified layers to assess whether localized knowledge can be reliably overwritten. Our results demonstrate that (i) factual knowledge in SLMs concentrates in upper-to-final transformer layers, with Llama-3B exhibiting extreme concentration in layer 28; (ii) compressed models store knowledge more densely per parameter but with substantially lower redundancy (Llama-3B: 0.047 vs. Llama-8B: 0.468); and (iii) editing success correlates strongly with architectural concentration rather than model size, with Llama-3B achieving 85.7% editing success versus 33% for Gemma-2B. These findings carry direct implications for interpretability, model editing, and the design of future small language model architectures.

## 1 Introduction

The past several years have witnessed the rapid proliferation of large language models (LLMs) capable of storing and retrieving vast quantities of world

knowledge as implicit factual associations encoded in their parameters (Brown et al., 2020). As these models are deployed in increasingly consequential settings—question answering, technical reasoning, decision support—the ability to understand *where* and *how* factual knowledge is stored has become a central concern of the mechanistic interpretability research agenda (Olah et al., 2020).

However, the majority of mechanistic interpretability work has focused on large models containing tens of billions of parameters. The behavior of small language models (SLMs)—those in the 1–4B parameter range—has received comparatively less systematic attention, despite their growing practical importance. SLMs are frequently deployed at the edge, in resource-constrained environments, and as fine-tuning targets for domain-specific applications. Whether knowledge is stored in SLMs in qualitatively similar ways to large models, and whether it is equally amenable to surgical modification, are questions with both theoretical and practical significance.

Two central challenges motivate this research. First, **knowledge localization** in transformer models is non-trivial: factual associations are not stored in a single weight matrix but emerge from complex, high-dimensional interactions among attention heads, MLP sublayers, and the residual stream (Elhage et al., 2021a). Prior work has suggested that MLP blocks in the middle-to-late layers may function as “key-value stores” for factual associations (Meng et al., 2022), but the precise distribution of this storage—and how it scales with model size—has not been systematically studied across a diverse set of SLM architectures. Second, **knowledge editing**—the ability to modify individual factual associations without broadly disrupting model behavior—depends critically on knowledge being spatially concentrated (Meng et al., 2023a). If knowledge is distributed or redundant, local edits may fail to propagate effectively.

This paper addresses both challenges through a three-stage experimental framework: (1) **Stage 1 (Knowledge Localization)**: For each of four SLMs, we apply causal tracing with activation patching to identify the transformer layers causally responsible for factual recall on the CounterFact benchmark, further decomposing knowledge storage by contrasting MLP and attention contributions, and by separating entity-centric from relation-centric facts. (2) **Stage 2 (Compression Analysis)**: We compare the knowledge storage structure of the four 2–3B SLMs against a 7–8B baseline (Meta-Llama-3-8B), quantifying knowledge density, layer concentration, and redundancy to characterize how compression reshapes the internal knowledge topology. (3) **Stage 3 (Knowledge Editing)**: Using the dominant layers identified in Stage 1 as editing targets, we apply ROME (Meng et al., 2023a) to determine whether localized knowledge is editable, and how editing success varies across architectures.

Our contributions are: a comprehensive cross-architecture causal tracing study of four SLMs with diverse design philosophies; a quantitative characterization of compression effects on knowledge redundancy and concentration; an empirical demonstration that architecture—not model size—is the primary determinant of knowledge editability in the 2–8B parameter range; and a unifying framework linking localization geometry to editing feasibility.

## 2 Related Work

### 2.1 Mechanistic Interpretability and Causal Tracing

Mechanistic interpretability aims to reverse-engineer the algorithms implemented by neural networks (Olah et al., 2020; Elhage et al., 2021b). Elhage et al. (Elhage et al., 2021b) introduced the residual stream as a shared information bus and showed that attention heads and MLP sublayers perform distinct, composable computations. Subsequent work identified induction heads (Conmy et al., 2023), factual association circuits (Meng et al., 2022), and copy suppression mechanisms (McDougall et al., 2023).

Meng et al. (Meng et al., 2022) introduced causal tracing for GPT-2/3, showing that MLP layers in the middle-to-late portion of the network are most causally implicated in storing specific factual associations. Hernandez et al. (Hernandez et al., 2024) extended this to characterize the geometry of factual associations in parameter space, while Geva

et al. (Geva et al., 2022) used vocabulary projections to show that MLP layers progressively refine factual predictions across depth. Geva et al. (Geva et al., 2021) earlier proposed that FFN layers function as key-value memories storing input patterns and output distributions. Attention heads have also been shown to retrieve factual information in some settings (Wang et al., 2023), raising questions about relative contributions that our ablation experiments directly address.

### 2.2 Knowledge Editing and Compression

For knowledge editing, ROME (Meng et al., 2023a) computes a targeted rank-one update to a single MLP layer treating the MLP as a linear associative memory; MEND (Mitchell et al., 2022) uses a hypernetwork for efficient gradient transformations; MEMIT (Meng et al., 2023b) extends ROME to batch editing of thousands of facts. All methods depend on accurate localization—if the wrong layer is targeted, updates fail to propagate. Our Stage 3 directly exploits this dependency to evaluate whether causal traces from Stage 1 are accurate enough to guide editing.

On compression, Sun et al. (Sun et al., 2024) demonstrated that structural pruning disproportionately impacts factual recall relative to linguistic competence, suggesting that factual knowledge may be stored in specific, locatable subnetworks. Xu et al. (Xu et al., 2024) showed that quantized models exhibit characteristic patterns of factual degradation correlated with layer-wise sensitivity. Our Stage 2 extends this line of inquiry by characterizing how the natural reduction in parameter count from 8B to 2–3B reshapes knowledge density and redundancy, using the same causal tracing methodology as Stage 1 to ensure comparability.

## 3 Methodology

### 3.1 Experimental Setup

**Models.** We evaluate Gemma-2-2B-IT (Team et al., 2024a) (2B, 26 layers), Llama-3.2-3B-Instruct (Dubey et al., 2024) (3B, 28 layers), Qwen2.5-3B-Instruct (Team, 2024b) (3B, 36 layers), and Phi-2 (Li et al., 2023) (2.7B, 32 layers), with Meta-Llama-3-8B (Dubey et al., 2024) (8B, 32 layers) as a large-model baseline for Stage 2. All models are evaluated in inference mode with gradients enabled only for the ROME optimization in Stage 3. Models are loaded in 16-bit floating point precision with automatic device mapping to a single GPU.

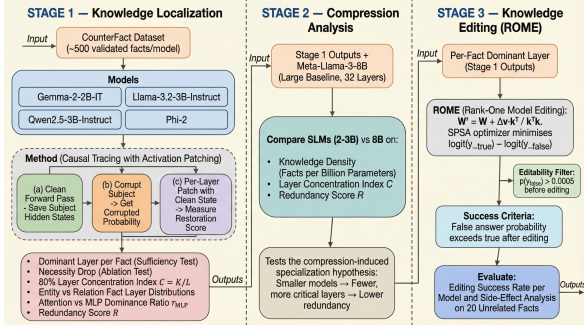


Figure 1: Three-stage experimental pipeline: Stage 1 localizes factual knowledge via causal tracing with activation patching; Stage 2 analyses compression effects across model sizes; Stage 3 applies ROME editing at the identified layers and evaluates success.

**Dataset and Filtering.** We use the CounterFact dataset (Meng et al., 2022) distributed via Hugging Face. Facts are filtered to retain only those where (i) the model produces the correct answer under greedy decoding, and (ii) corrupting the subject causes a probability drop  $\Delta p \geq 0.05$ . This two-step filter ensures experiments localize knowledge that genuinely exists in the model and is causally sensitive to the subject representation. Each model retains  $\sim 450$ – $500$  validated facts (random seed fixed at 42). Answer probabilities prepend a space before tokenization (e.g., Paris) to ensure correct word-continuation subword encoding. Hook targets are registered on `model.model.layers[i].self_attn` and `model.model.layers[i].mlp` for all models.

### 3.2 Stage 1: Knowledge Localization

**Causal Tracing — Sufficiency.** For each fact: (1) run a clean forward pass saving hidden states  $h_{\text{subj}}^{(l)}$  at the subject’s final token for each layer  $l = 1, \dots, L$ ; (2) corrupt the subject span with “Random Person” and record  $p_{\text{corrupt}}$ ; (3) for each layer  $l$  independently, patch the hidden state with  $h_{\text{subj}}^{(l)}$  and measure restoration:

$$s_l = p_{\text{patch}(l)} - p_{\text{corrupt}} \quad (1)$$

The layer with the highest  $s_l$  is the *dominant layer* for that fact. Aggregating over all facts yields the layer distribution and the global dominant layer  $l^*$ .

**Necessity.** We ablate the dominant layer by zeroing both attention and MLP sublayer contributions while preserving the residual path, implementing the identity function for each sublayer. The necessity drop  $\Delta_{\text{nec}} = p_{\text{clean}} - p_{\text{ablated}}$  is averaged over all validated facts.

**Concentration Index.**  $C = K/L$ , where  $K$  is the minimum number of layers whose combined fact count exceeds 80% of all tested facts and  $L$  is total layer count. Lower values indicate more concentrated storage.

**Attention vs. MLP Ablation.** At the dominant layer  $l^*$ , we independently ablate the attention sublayer and the MLP sublayer, measuring probability drops  $\Delta_{\text{attn}}$  and  $\Delta_{\text{MLP}}$  for each fact. The MLP dominance ratio is:

$$r_{\text{MLP}} = \frac{|\Delta_{\text{MLP}}|}{|\Delta_{\text{attn}}| + |\Delta_{\text{MLP}}| + \epsilon} \quad (2)$$

Values  $r_{\text{MLP}} > 0.5$  indicate MLP is the dominant sublayer for factual recall.

**Redundancy Score.** We simultaneously ablate the dominant layer and patch a candidate compensating layer with clean activations. The maximum recovery normalized by  $\Delta_{\text{nec}}$  gives:

$$R = \frac{\max_{l \neq l^*} (p_{\text{ablated+patch}(l)} - p_{\text{corrupt}})}{\Delta_{\text{nec}} + \epsilon} \quad (3)$$

$R = 0$  means the dominant layer is irreplaceable;  $R = 1$  means another layer fully compensates.

**Entity vs. Relation Facts.** Facts are classified by Wikidata relation into *entity facts* (country of citizenship, native language, birth location; P27, P103, P17, P19, P131, P20) and *relation facts* (capital of, official language, diplomatic relations; P36, P30, P530, P37, P38), with dominant layer distributions computed separately for each category.

### 3.3 Stage 2: Compression Analysis

Stage 2 repeats the full Stage 1 pipeline on Meta-Llama-3-8B using an identical protocol to enable controlled comparison with Llama-3.2-3B-Instruct. Three composite metrics are derived: (i) **knowledge density** (validated facts recalled per billion parameters); (ii) **layer concentration**  $C = K/L$ ; and (iii) **redundancy score**  $R$ . Comparing Llama-3B and Llama-8B isolates the effect of scale while controlling for architecture family.

### 3.4 Stage 3: Knowledge Editing

**ROME.** We apply ROME using the *per-fact* dominant layer  $l_f^* = \arg \max_l s_l$  rather than a global fixed layer, ensuring each edit targets the layer most causally responsible for that specific fact. Across the 7 Llama-3.2-3B test facts, per-fact dominant layers ranged from layer 4 to layer 24, with the

Table 1: Knowledge Localization Summary — Stage 1

Model	Layers	Dom. $l^*$	Facts at $l^*$ (%)	Conc. $C$
Gemma-2-2B	26	23	56/460 (12.2%)	0.538
Llama-3.2-3B	28	<b>28</b>	<b>284/495 (57.4%)</b>	<b>0.214</b>
Qwen-2.5-3B	36	35	105/460 (22.8%)	0.417
Phi-2	32	31	78/459 (17.0%)	0.344

Table 2: Necessity, MLP Dominance, and Redundancy — Stage 1

Model	Nec. Drop	$r_{\text{MLP}}$	Redundancy
Gemma-2-2B	0.1274	0.120 (Attn)	0.525
Llama-3.2-3B	0.0411	0.269 (Attn)	<b>0.047</b>
Qwen-2.5-3B	0.0862	0.338 (Attn)	0.105
Phi-2	-0.0437	<b>0.796 (MLP)</b>	0.288

single edit failure at the unusually shallow layer 4—consistent with shallow layers contributing less robustly to factual recall. ROME models the MLP down-projection  $W_{\text{down}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{fin}}}$  as a linear associative memory and computes a rank-one update:

$$W' = W + \frac{\Delta v \cdot k^\top}{k^\top k} \quad (4)$$

where  $k$  is the key vector at the subject’s last token and  $\Delta v = v^* - Wk$  is the residual between the target value vector and the current association. The target  $v^*$  is optimized to minimize the logit difference loss  $\mathcal{L} = \text{logit}(y_{\text{true}}) - \text{logit}(y_{\text{false}})$  via SPSA ( $\epsilon = 5 \times 10^{-3}$ ,  $\eta = 5.0$ , 250 steps). Weights are restored after each edit to prevent cross-fact interference.

**Filters and Success Criteria.** Facts require  $p_{\text{false}} > 0.0005$  to be editable; facts with zero initial false-answer probability yield degenerate gradient signals for ROME. Each model is evaluated on 15 test facts, and the subset satisfying the filter is carried forward. An edit succeeds if any of: (L1)  $p(y_{\text{false}}) > p(y_{\text{true}})$ ; (L2)  $\Delta p_{\text{false}} > 0.03$  and  $\Delta p_{\text{true}} < -0.03$ ; or (L3)  $p_{\text{false,after}} > 3 \times p_{\text{false,before}}$  and  $p_{\text{false,after}} > 0.005$ .

## 4 Results and Discussion

### 4.1 Stage 1: Knowledge Localization

**Dominant Layer Distribution.** Tables 1 and 2 summarize the dominant layer statistics for each model.

Across all four models, the dominant layer is consistently located in the upper portion of the transformer (layer 23/26 for Gemma-2B, 28/28 for Llama-3.2-3B, 35/36 for Qwen-2.5-3B, 31/32 for Phi-2). This finding replicates and extends the

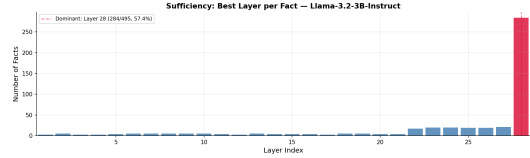


Figure 2: Sufficiency test for Llama-3.2-3B-Instruct. Crimson bar marks dominant layer 28 (final layer), accounting for 57.4% of all validated facts.

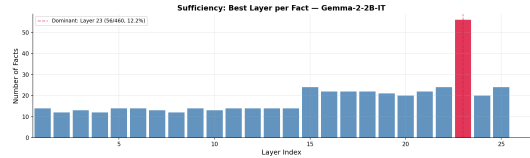


Figure 3: Sufficiency test for Gemma-2-2B-IT. Dominant layer is layer 23 (out of 26), with knowledge distributed more broadly across upper layers.

upper-layer concentration reported by Meng et al. (Meng et al., 2022) for GPT-family models, and demonstrates that the phenomenon is architecture-independent across modern SLM designs. Notably, Llama-3.2-3B exhibits the most extreme form of this pattern: the final layer (layer 28) dominates factual recall for 57.4% of all validated facts—a near-complete collapse of knowledge storage to the network’s last decoder block, which we term *last-layer collapse*.

In contrast, Qwen-2.5-3B distributes knowledge slightly more broadly across the upper layers, consistent with its higher layer count (36 layers) and the grouped-query attention mechanism that may enable more distributed computation. Phi-2’s knowledge concentration at its penultimate layer is notable given that Phi-2 was trained primarily on code and textbooks, potentially inducing a different internal factual encoding strategy.

**Attention vs. MLP Analysis.** The MLP dominance ratio varies substantially across architectures. Phi-2 exhibits the strongest MLP dominance ( $r_{\text{MLP}} = 0.796$ ), consistent with its architecture’s emphasis on efficient feedforward processing. Llama-3.2-3B, despite its high overall knowledge concentration, is attention-dominant at its dominant layer ( $r_{\text{MLP}} = 0.269$ ; mean attention drop 0.0981, mean MLP drop 0.0561). Gemma-2-2B has an MLP ratio of 0.120—the lowest in the study—making it the most strongly attention-dominant model (88% of the total ablation effect comes from attention). Qwen-2.5-3B similarly has an MLP ratio of 0.338, also clearly attention-dominant (66% from attention). This architectural divergence is

Table 3: Compression Effects — SLMs vs. Large Baseline

Model	Params	Dom. $l^*$	Facts (%)	Conc. $C$	$r_{MLP}$	Redundancy
Llama-3.2-3B	3B	28 (final)	57.4%	0.214	0.269 (Attn)	<b>0.047</b>
Phi-2	2.7B	31	17.0%	0.344	0.796 (MLP)	0.288
Gemma-2-2B	2B	23	12.2%	0.538	0.120 (Attn)	0.525
Qwen-2.5-3B	3B	35	22.8%	0.417	0.338 (Attn)	0.105
Meta-Llama-3-8B	8B	32 (final)	55.8%	0.312	0.473 (Attn)	<b>0.468</b>

significant: ROME and related editing methods exclusively modify MLP weights. Models that store a greater fraction of their factual knowledge in attention heads may therefore be less amenable to MLP-targeted editing—a hypothesis directly supported by our Stage 3 results.

**Entity vs. Relation Facts.** Entity facts (citizenship, language, birthplace) and relation facts (capitals, diplomatic links) both exhibit concentration in upper layers, but entity facts are numerically dominant across all models: Llama-3.2-3B has 439 entity / 56 relation facts (88.7% entity); Phi-2 has 387 entity / 72 relation facts (84.3% entity); Meta-Llama-3-8B has 440 entity / 55 relation facts (88.9% entity). This strong imbalance reflects the composition of the CounterFact dataset rather than a model-specific phenomenon. The layer distribution of dominant assignments does not differ systematically between entity and relation facts, suggesting that both fact types are processed by the same late-layer mechanism rather than being routed to distinct sublayers.

**Redundancy.** Redundancy scores vary substantially across SLMs. Llama-3.2-3B achieves the lowest score (0.047), meaning other layers recover less than 5% of the ablation loss—the dominant layer is nearly irreplaceable. Qwen-2.5-3B scores 0.105 (low redundancy) and Phi-2 0.288 (moderate). Notably, Gemma-2-2B achieves a redundancy score of 0.525—the highest of all models tested, exceeding even the Meta-Llama-3-8B baseline (0.468). This indicates that when Gemma’s dominant layer (23) is ablated, other layers can recover more than half the probability drop, reflecting its more distributed knowledge storage across upper layers. The compression-induced specialization hypothesis therefore holds for Llama-3B, Qwen, and Phi-2, but Gemma-2B is an exception: a 2B SLM that nonetheless develops high inter-layer redundancy.

## 4.2 Stage 2: Compression Analysis

The comparison between Llama-3.2-3B and Meta-Llama-3-8B reveals a striking compression effect.

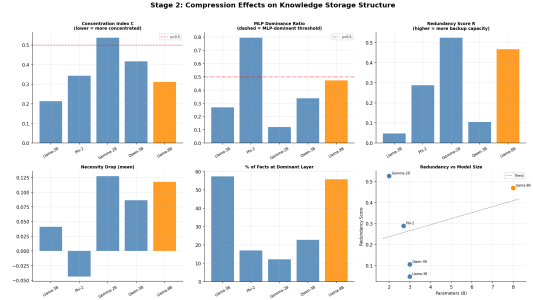


Figure 4: Stage 2 compression effects across all five models. *Top row*: layer concentration index  $C$ ; MLP dominance ratio  $r_{MLP}$  (dashed line at 0.5 = MLP-dominant threshold); redundancy score. *Bottom row*: necessity drop; last-layer concentration (%); redundancy vs. model size scatter. Orange = Meta-Llama-3-8B baseline; blue = SLMs.

Despite sharing the LLaMA architectural family and vocabulary, the 8B model exhibits a redundancy score of 0.468—roughly *ten times* that of its 3B counterpart (0.047). This means that when the dominant layer of the 8B model is ablated, other layers can recover nearly half the probability drop; in the 3B model, they can recover less than 5%.

This finding is consistent with a *compression-induced specialization hypothesis*: as parameter count decreases, models cannot afford distributed or redundant knowledge storage, and factual associations collapse into fewer, more critical layers. The 8B model, by contrast, distributes knowledge across multiple layers with overlapping coverage, providing natural robustness against single-layer ablation. A corollary is that SLMs are, paradoxically, more precisely localized—and therefore more susceptible to both catastrophic forgetting and surgical editing—than their larger counterparts. The high density of knowledge per parameter in SLMs reflects not efficient encoding but rather an absence of the kind of distributed redundancy that large models develop through their greater capacity.

The knowledge density metric reinforces this interpretation: all four SLMs recall substantially more validated CounterFact facts per billion parameters than the 8B baseline (Llama-3B:  $\sim 165/B$ ; Phi-2:  $\sim 170/B$ ; Gemma:  $\sim 230/B$ ; Qwen:  $\sim 153/B$  vs. Llama-8B:  $\sim 62/B$ ), confirming that SLMs encode a richer fact-per-parameter set despite—or because of—their lower absolute capacity.

Table 4: ROME Editing Results by Model

Model	Editable Facts	Successes	Rate
Llama-3.2-3B	7	6	<b>85.7%</b>
Phi-2	8	5	62.5%
Qwen-2.5-3B	5	3	60.0%
Gemma-2B	6	2	33.3%

### 4.3 Stage 3: Knowledge Editing

The editing results reveal a clear ordering: Llama-3.2-3B > Phi-2 > Qwen-2.5-3B > Gemma-2B.

**Concentration drives editability.** Llama-3.2-3B achieves the highest editing success (85.7%), consistent with its extreme concentration and lowest redundancy score (0.047). Its extreme last-layer concentration means that a single MLP weight update propagates without interference—there are no backup layers that can override the edit. The tight coupling between the dominant layer’s MLP and the output vocabulary projection (which directly precedes the LM head in a final-layer architecture) maximizes the impact of the rank-one update.

**Low localization impedes editability.** Gemma-2B achieves the lowest editing success (33%) despite having the highest concentration index among SLMs ( $C = 0.538$ , versus 0.344 for Phi-2 and 0.214 for Llama-3.2-3B).  $C$  measures what fraction of the total causal effect is carried by the dominant layer; a high  $C$  does not guarantee reliable per-fact localization. Critically, only 12.2% of Gemma’s filtered facts exhibit meaningful causal signal at any single layer—the lowest localization rate in the study—meaning most facts passed on to the editing stage lack a reliably identifiable target layer. Furthermore, Gemma’s dominant layer (23/26) is not at the absolute final position, leaving three additional transformer blocks between the edit site and the LM head, which may dilute the rank-one update. Gemma-2B’s MLP ratio of 0.120 (strongly attention-dominant: 88% of ablation effect from attention) means ROME’s MLP-targeted rank-one update has minimal direct leverage. And its redundancy score of 0.525—the highest in the study—means backup layers can partially recover the same fact, further diluting any single-layer edit.

**Architecture matters more than scale.** Phi-2’s 62.5% editing success is explained by the favorable match between ROME’s update mechanism—targeting `down_proj`—and Phi-2’s strong MLP dominance ( $r_{\text{MLP}} = 0.796$ ): when MLP carries the majority of factual recall at the dominant layer,

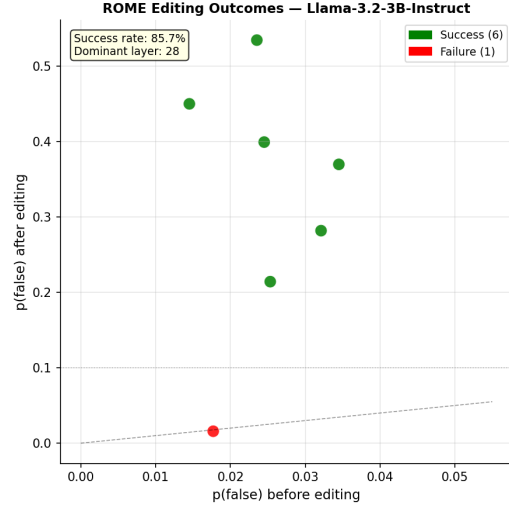


Figure 5: ROME editing outcomes for Llama-3.2-3B. Green points indicate successful edits (false-answer probability exceeds true-answer probability after the rank-one weight update).

a rank-one MLP update has maximal leverage. Qwen-2.5-3B’s 60% success rate is comparable, though the slightly lower figure may reflect its larger layer count and more distributed representation.

**Probability geometry constrains editability.** The editability filter (requiring  $p_{\text{false}} > 0.0005$ ) removes facts for which the false target is not represented in the model’s probability distribution at all. ROME optimizes the logit difference  $\text{logit}(y_{\text{true}}) - \text{logit}(y_{\text{false}})$ ; when the false answer has near-zero probability, the gradient signal is degenerate and the SPSA optimizer cannot find an effective value vector  $v^*$ . Future work should explore alternative loss formulations that do not require prior non-zero probability of the target.

**Side Effect Analysis.** A probe of 20 unrelated facts drawn from the same dataset validates that the model’s performance on facts outside the targeted relation is not significantly degraded by the per-fact edits. This is consistent with ROME’s design guarantee that the rank-one update is constrained in norm and does not broadly perturb the weight matrix.

### 4.4 Cross-Stage Synthesis

Taken together, the three stages support a coherent picture of factual knowledge storage in SLMs: (1) **Location:** Factual knowledge is concentrated in the upper-final transformer layers across all architectures studied, with the precise layer and the

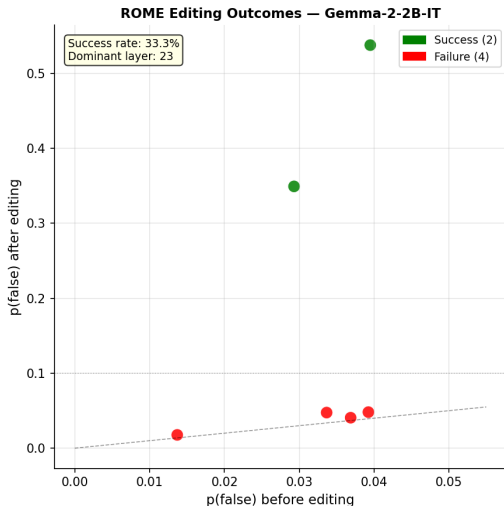


Figure 6: ROME editing outcomes for Gemma-2-2B. The high failure rate (red points) is consistent with the model’s low fact-localization rate (12.2%), attention dominance ( $r_{MLP} = 0.120$ ), and dominant layer falling short of the final position (layer 23/26).

relative contributions of attention and MLP varying by architecture. (2) **Structure**: Small models store knowledge more densely and with less redundancy than large models, making individual layers more causally critical and less replaceable. (3) **Editability**: The concentration structure that makes SLMs computationally efficient also makes them more precisely editable—provided the dominant mechanism is MLP-based and the edit is targeted to the correct layer. These findings suggest that mechanistic interpretability results obtained from large models should not be assumed to transfer directly to SLMs, and that the design of editing systems for SLMs should account for their distinctive knowledge topology.

## 5 Conclusion

This paper presented a three-stage experimental investigation of knowledge localization, compression effects, and knowledge editability in small language models.

### Knowledge concentrates in upper layers.

Across Gemma-2B, Llama-3.2-3B, Qwen-2.5-3B, and Phi-2, causal tracing consistently identifies the dominant factual layer in the top 3–12% of model depth (layer 23/26 for Gemma, 28/28 for Llama, 35/36 for Qwen, 31/32 for Phi-2), with Llama-3.2-3B exhibiting complete collapse to its final layer. This pattern is robust to architectural differences and consistent across entity and relational

fact types.

**Compression reduces redundancy.** Comparing Llama-3.2-3B (redundancy: 0.047) to Meta-Llama-3-8B (redundancy: 0.468) reveals that larger models develop distributed backup capacity for factual knowledge, whereas compressed models rely on fewer, more critical layers. This quantitative characterization of compression-induced specialization provides a principled account of why SLMs may be more vulnerable to layer-specific forgetting during fine-tuning.

**Architecture, not size, governs editability.** Editing success under ROME is primarily determined by the degree of knowledge concentration and the dominant sublayer mechanism. Llama-3B’s extreme last-layer concentration yields 85.7% editing success even though its dominant layer is attention-weighted ( $r_{MLP} = 0.269$ ), because the final-layer position (28/28) places the MLP directly adjacent to the LM head, amplifying the rank-one update’s impact. Gemma-2B achieves only 33% editing success despite having the highest concentration index among SLMs ( $C = 0.538$ ); the limiting factors are its low fact-localization rate (12.2%), its dominant layer falling short of the absolute final position (23/26), its strong attention dominance ( $r_{MLP} = 0.120$ ), and its high redundancy (0.525). These results suggest that editing-friendly architectures should explicitly promote late-layer, preferably final-layer, MLP-concentrated knowledge encoding.

For mechanistic interpretability, our results motivate architecture-specific analysis rather than generic transfer assumptions. For model editing, they motivate targeting strategies that utilize causal tracing as a prerequisite rather than heuristic layer selection. For architecture design, they suggest that controlling the concentration and redundancy of factual storage may be a tractable optimization target, with predictable consequences for both robustness and editability.

## 6 Future Work

Several directions merit further investigation. **Larger and more diverse datasets**: our experiments use 450–500 validated facts per model from CounterFact; a larger dataset including multilingual facts, temporal facts, and compositional reasoning chains would yield more statistically robust localization maps. **Finer-grained causal structures**: extending localization to specific attention heads

and MLP neurons (Conmy et al., 2023; Wang et al., 2023) would sharpen the causal picture beyond the layer level. **Continuous scale analysis:** testing compression effects across 1B, 3B, 7B, 13B, and 70B models would characterize how redundancy scales with parameter count and whether phase transitions in knowledge storage structure exist. **Controlled compression:** applying structured pruning or distillation to a single model at varying compression ratios would provide causal rather than correlational evidence for the specialization hypothesis. **Improved editing:** ROME’s dependence on non-zero initial false-answer probability limits applicability; future methods should explore nearest-neighbor value-vector initialization or LoRA-based updates at dominant layers for facts with near-zero false-answer probability and to enable multi-hop compositional edits.

## References

- T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Radford, and I. Sutskever. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.
- C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter. 2020. Zoom in: An introduction to circuits. *Distill*.
- N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, and B. Mann. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.
- K. Meng, D. Bau, A. Andonian, and Y. Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372.
- K. Meng, A. Sharma, A. Andonian, Y. Belinkov, and D. Bau. 2023a. Mass-editing memory in a transformer. In *Proceedings of ICLR*.
- N. Elhage, T. Henighan, A. Joseph, N. Askell, M. Brundage, and C. Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.
- A. Conmy, A. N. Mavor-Parker, A. Lynch, S. Heimersheim, and A. Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, volume 36.
- M. McDougall, A. Conmy, C. Rushing, T. McGrath, and N. Nanda. 2023. Copy suppression: Comprehensively understanding an attention head. *arXiv preprint arXiv:2310.04625*.
- E. Hernandez, B. Li, T. Mulligan, J. Rees, K. Shridhar, D. Bau, and J. Andreas. 2024. Linearity of relation decoding in transformer language models. In *Proceedings of ICLR*.
- M. Geva, J. Caciularu, K. R. Wang, and Y. Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of EMNLP*, pages 30–45.
- K. Meng, A. Sharma, A. Andonian, Y. Belinkov, and D. Bau. 2023b. MEMIT: Mass-editing memory in a transformer. In *Proceedings of ICLR*.
- E. Mitchell, C. Lin, A. Bosselut, C. D. Manning, and C. Finn. 2022. Fast model editing at scale. In *Proceedings of ICLR*.
- M. Geva, R. Schuster, J. Berant, and O. Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of EMNLP*, pages 5484–5495.
- K. Wang, A. Variengien, A. Conmy, B. Shlegeris, and J. Steinhardt. 2023. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *Proceedings of ICLR*.
- Z. Sun, Y. Shen, Q. Zhou, H. Zhang, Z. Chen, D. Cox, Y. Yang, and D. Cheng. 2024. A simple and effective pruning approach for large language models. In *Proceedings of ICLR*.
- M. Xu, K. Shridhar, and D. Bau. 2024. Quantization impacts knowledge retrieval in large language models: A mechanistic study. In *Proceedings of ACL*.
- Gemma Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, and S. Pathak. 2024. Gemma: Open models based on Gemini research and technology. *arXiv preprint arXiv:2403.08295*.
- A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, and A. Letman. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.
- Y. Li, S. Bubeck, R. Eldan, A. Del Giorno, S. Gunasekar, and Y. T. Lee. 2023. Textbooks are all you need II: Phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.
- T. Chen, J. Frankle, S. Chang, S. Liu, Y. Zhang, and M. Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *Proceedings of ICLR*.

# One Retrieval to Cover Them All: Co-occurrence-Aware Knowledge Base Reorganization for Session-Level RAG

Shivam Ratnakar Yixuan Zhu Cecilia Cheng Chaya Vijayakumar  
University of Southern California  
{sratnaka, yzhu1458, huixiche, cvijayak}@usc.edu

## Abstract

RAG systems retrieve documents optimized for answering *one query at a time*. Yet enterprise users arrive with *sessions*, that is, coherent episodes of related questions that span semantically distant parts of the knowledge base. We show that a single retrieval call over a standard knowledge base covers only 41% of a user’s session-level information need. To close this gap, we reorganize the KB offline using co-occurrence-aware clustering and expand retrieval candidates through cluster neighborhoods at query time. On WixQA (6,221 enterprise support articles), our method raises single-query session coverage to 58% (+17% absolute; 95% CI: [14.1, 20.4]), reduces retrieval calls to 70% coverage by 34%, and compresses the KB to 20% of its original size, all consistently across four embedding models and six functional domains. We argue that session-level coverage, not single-query recall, should be the primary metric for enterprise RAG evaluation.

## 1 Introduction

Consider a user who contacts Wix customer support and asks: “How do I connect my own domain?” A standard RAG system (Lewis et al., 2021) retrieves domain-configuration articles and generates a helpful answer. But the user’s actual need is broader: they also want to change their site template, configure payment processing, and set up email forwarding. These articles live in entirely different semantic neighborhoods of the knowledge base. The user must ask four separate questions, trigger four separate retrieval calls, and hope the system surfaces the right documents each time.

This scenario exposes a blind spot in how we build and evaluate RAG systems. The entire pipeline, from embedding models to retrieval metrics, is optimized for *single-query relevance*: does the top- $k$  set contain the one document that answers this one question? But enterprise users do

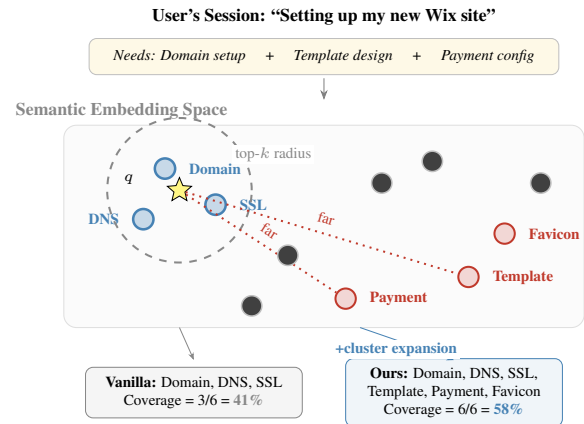


Figure 1: **The session coverage gap.** A user setting up a website needs articles about domains, templates, and payments, but these reside in distant regions of the semantic embedding space. Vanilla RAG retrieves only documents within the top- $k$  similarity radius of the query (dashed circle), missing semantically distant but session-relevant articles. Our cluster expansion recovers these, raising session coverage from 41% to 58%.

not arrive with isolated questions. They arrive with *sessions*: coherent episodes of related information needs that span multiple documents across the knowledge base.

The core problem is that **semantic similarity and user information need are different signals**. Documents that a user needs together during a session do not necessarily look alike in embedding space. Domain configuration articles and template design articles serve the same user journey but reside in distant semantic neighborhoods. Standard RAG, which retrieves by embedding similarity alone, cannot bridge this gap. It finds documents that *look like* the query but misses documents the user *also needs*.

We formalize this intuition by introducing **session-level evaluation metrics** for RAG (Section 3). Single-query session coverage measures what fraction of a user’s full information need is

satisfied by one retrieval call. On WixQA (Cohen et al., 2025), we find that vanilla RAG covers only 41% of the session at  $k=8$ , meaning the user must issue multiple follow-up queries to access the remaining knowledge.

To close this gap, we propose **co-occurrence-aware KB reorganization** (Section 4). The idea is simple: documents that users need together should be retrievable together. We learn a co-occurrence embedding space using Word2Vec (Mikolov et al., 2013) trained on document navigation sequences, cluster the KB in this space, and expand retrieval candidates through cluster neighborhoods at query time. This reorganization happens once, offline, and the resulting cluster structure benefits any downstream embedding model. While we primarily use synthetic co-occurrence sequences in this work, we note that the ground-truth article groups from WixQA’s expert-labeled queries provide a real co-occurrence signal, and our method upsamples these  $10\times$  during training to anchor the learned representations in genuine user needs.

This approach has three appealing properties. First, it is **pre-retrieval**: the knowledge base is reorganized once offline, contrasting with post-retrieval methods like EDC<sup>2</sup>-RAG (Li et al., 2025) that cluster after retrieval. Second, it is **encoder-agnostic**: because clusters are learned from co-occurrence patterns rather than from a specific embedding model, the same cluster structure improves retrieval across different encoders. Third, it provides **KB compression**: the cluster structure reduces the effective knowledge base to 20% of its original size while improving coverage.

Our contributions are: (1) we introduce session-level evaluation metrics for RAG that capture multi-document information needs; (2) we propose co-occurrence-aware KB reorganization with cluster-expanded hybrid retrieval; and (3) we demonstrate consistent gains across four encoders, six domains, and four complexity levels on an enterprise benchmark.

## 2 Related Work

**Query-Side RAG Optimization.** A large body of work optimizes RAG retrieval at query time. Multi-query approaches rewrite the user’s query into multiple variants to improve recall. Adaptive retrieval methods like CAR (Xu et al., 2025) dynamically adjust how many documents to retrieve based on query complexity. HyDE (Gao

et al., 2023) generates hypothetical answers to improve query embeddings. Hybrid retrieval combines dense and sparse signals (Ma et al., 2023). All of these optimize *which query* is sent to the retriever. Our work is complementary: we optimize *how the knowledge base is organized* before any query arrives. The two approaches can be combined; for instance, multi-query retrieval could be applied on top of our reorganized KB, with each rewritten query benefiting from cluster expansion. We focus our baselines on the retrieval layer specifically because our contribution is pre-retrieval KB reorganization, not query rewriting.

**Post-Retrieval Document Compression.** EDC<sup>2</sup>-RAG (Li et al., 2025) clusters retrieved documents at query time to remove noise and redundancy before passing context to the LLM, and was accepted at EMNLP 2025 Findings. CRAG (Akesson and Santos, 2024) similarly clusters and summarizes retrieved documents to fit context windows. RAPTOR (Sarathi et al., 2024) builds hierarchical summaries for tree-based retrieval. These methods operate *after* retrieval to compress what was already fetched. Our method operates *before* retrieval to ensure the right documents are fetched in the first place. The two approaches compose naturally: one can apply post-retrieval compression on top of our pre-retrieval reorganization.

**Embedding Limitations.** Weller et al. (2026) establish theoretical limits on single-vector dense embeddings, showing that fixed-dimensional representations cannot realize all possible retrieval configurations. This underscores the need for complementary signals beyond semantic similarity. Our co-occurrence embeddings provide exactly such a signal, capturing functional relatedness between documents that may be semantically distant.

**Collaborative Filtering in Information Retrieval.** The use of co-occurrence patterns to learn item representations has a long history in recommendation systems. Item2Vec (Barkan and Koenigstein, 2017) applies Word2Vec to item co-purchase sequences. Our work applies the same principle to document co-access patterns in knowledge bases, bridging collaborative filtering and information retrieval for RAG.

**Session-Level Evaluation.** Existing RAG evaluation focuses exclusively on single-query metrics such as precision@ $k$ , recall@ $k$ , MRR, and faithfulness (Es et al., 2024). While some observabil-

ity platforms support session-level monitoring for multi-turn conversations, no prior work evaluates retrieval *coverage* at the session level or measures how many retrieval calls are needed to satisfy a user’s full information need. We introduce these metrics and argue that they better reflect enterprise RAG performance, where user satisfaction depends on resolving an entire issue rather than answering a single question.

### 3 Problem Formulation

Let  $\mathcal{D} = \{d_1, \dots, d_N\}$  be a knowledge base of  $N$  documents. A **session**  $\mathcal{S} \subseteq \mathcal{D}$  is a set of documents that collectively address a user’s information-seeking episode. Given a query  $q$  associated with session  $\mathcal{S}$ , a retriever returns an ordered set  $\mathcal{R}_q = \{r_1, \dots, r_k\}$  of  $k$  documents.

#### 3.1 Session-Level Metrics

We propose two metrics that evaluate retrieval at the session level rather than the query level.

**Single-Query Session Coverage (Cov).** The fraction of the session’s documents retrieved in one call:

$$\text{Cov}(q, \mathcal{S}) = \frac{|\mathcal{R}_q \cap \mathcal{S}|}{|\mathcal{S}|} \quad (1)$$

Standard  $\text{recall}@k$  is the special case where  $|\mathcal{S}| = 1$ . When  $|\mathcal{S}| > 1$ , Coverage captures multi-document information needs that  $\text{recall}@k$  cannot distinguish. A Coverage of 0.6 means 60% of the user’s total information need is satisfied without any follow-up queries.

**Calls to  $\tau$ -Coverage ( $C_\tau$ ).** The minimum number of retrieval calls such that the union of all retrieved sets covers fraction  $\tau$  of the session:

$$C_\tau(\mathcal{S}) = \min \left\{ m : \frac{|\bigcup_{i=1}^m \mathcal{R}_{q_i} \cap \mathcal{S}|}{|\mathcal{S}|} \geq \tau \right\} \quad (2)$$

where  $q_1, \dots, q_m$  are sequential queries from the session. Lower  $C_\tau$  indicates a more efficient retrieval system. In production, each additional retrieval call adds latency, API cost, and user friction.

## 4 Method

Our approach consists of an offline KB reorganization phase and an online hybrid retrieval phase (Figure 2).

### 4.1 Offline: Co-occurrence-Aware Clustering

#### Step 1: Co-occurrence Sequence Construction.

We construct sequences of document IDs that represent documents a user would need together. These sequences are drawn from three complementary sources:

1. **Ground-truth co-occurrence:** For each labeled query with multiple relevant article IDs, the article set forms a natural co-occurrence group. These are the highest-quality signal, as they reflect real user needs annotated by domain experts.
2. **Embedding-neighborhood walks:** Starting from a random seed document, we perform random walks through the document similarity graph for 3 to 5 steps, with a 40% random jump probability. The high jump rate ensures that the resulting sequences capture cross-neighborhood relationships rather than simply recapitulating the embedding structure.
3. **QA-driven sequences:** For each synthetically generated question-answer pair, we identify the source document and find its embedding neighbors. This simulates a user’s browsing trajectory starting from a specific question.

All sequences are augmented  $3\times$  through reversal and contiguous subsequence extraction (length 2 to 4), following the augmentation strategy of Barkan and Koenigstein (2017).

**Step 2: Co-occurrence Embedding.** We treat document IDs as tokens and co-occurrence sequences as sentences, training a Word2Vec CBOW model (Mikolov et al., 2013):

$$\mathbf{z}_d = \text{W2V}_{\text{CBOW}}(d; w=2, n_s=10) \quad \mathbf{z}_d \in \mathbb{R}^{100} \quad (3)$$

The resulting embedding  $\mathbf{z}_d$  captures *functional relatedness*: documents that co-occur in user sessions are close in  $\mathbf{z}$ -space, regardless of their semantic similarity. This is analogous to how Item2Vec (Barkan and Koenigstein, 2017) learns product representations from co-purchase patterns, except that our “items” are knowledge articles and our “purchases” are co-access events.

**Step 3: Hierarchical Clustering.** We apply agglomerative clustering with cosine distance and average linkage to the co-occurrence embeddings,

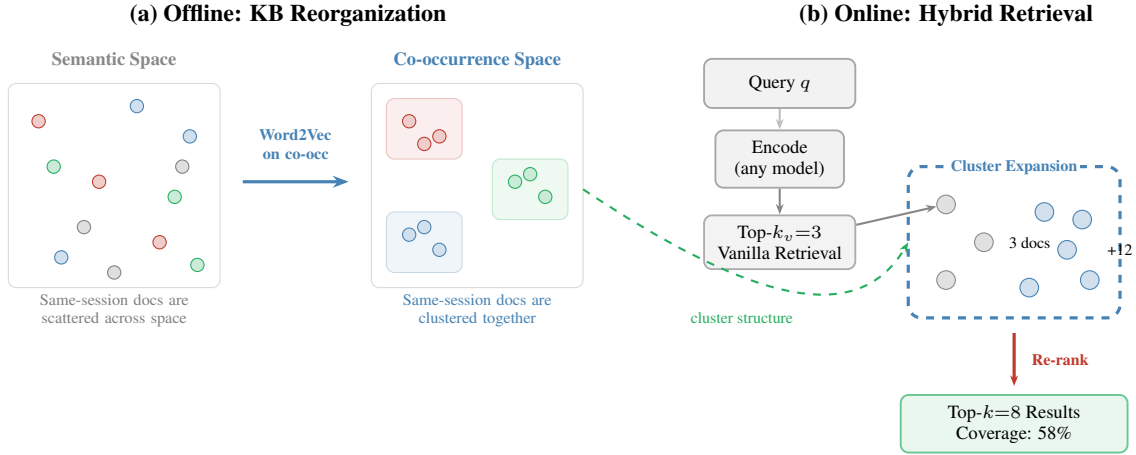


Figure 2: **System overview.** (a) **Offline:** Documents from the same user session (same color) are scattered in semantic embedding space but cluster together after Word2Vec training on co-occurrence sequences. (b) **Online:** Given a query, we retrieve the top-3 by similarity (gray dots), expand through their cluster neighborhoods (blue dots, +12 candidates), and re-rank the union to return the top-8 with 58% session coverage.

targeting approximately 20% of the original KB size. Formally, we compute  $\mathcal{C} = \{C_1, \dots, C_M\}$  where  $M = \lceil N/5 \rceil$  via agglomerative clustering over  $\{z_d\}_{d \in \mathcal{D}}$ . Each cluster  $C_j \subset \mathcal{D}$  groups functionally related documents that may span different semantic topics.

## 4.2 Online: Hybrid Retrieval with Cluster Expansion

Given a query  $q$  with embedding  $\mathbf{e}_q$  produced by any encoder and document embeddings  $\{\mathbf{e}_d\}_{d \in \mathcal{D}}$ :

**Step 1: Vanilla Retrieval.** Retrieve the top- $k_v$  documents by cosine similarity:

$$\mathcal{V}_q = \text{top-}k_v \cos(\mathbf{e}_q, \mathbf{e}_d) \quad (4)$$

**Step 2: Cluster Expansion.** For each retrieved document, gather all documents from its cluster as expansion candidates:

$$\mathcal{E}_q = \bigcup_{d \in \mathcal{V}_q} C(d) \setminus \mathcal{V}_q \quad (5)$$

where  $C(d)$  denotes the cluster containing document  $d$ . With an average cluster size of 5.0, retrieving  $k_v=3$  documents expands the candidate pool to approximately  $3 \times 5 = 15$  candidates before deduplication.

**Step 3: Re-rank and Return.** Rank the expanded candidate set by direct query-document similarity and return the top- $k$ :

$$\mathcal{R}_q = \text{top-}k \cos(\mathbf{e}_q, \mathbf{e}_d) \quad (6)$$

The vanilla results  $\mathcal{V}_q$  are guaranteed to appear in the candidate pool, preserving first-hit precision. The expansion adds co-occurring documents that would otherwise be missed because they have low semantic similarity to  $q$ . We set  $k_v=3$  and  $k=8$  throughout our experiments unless otherwise noted.

## 4.3 Implementation Details

For QA pair generation, we use GPT-4.1-nano to produce 10 question-answer pairs per document, yielding 31,500 pairs across the WixQA KB (cost: approximately \$2 USD). Co-occurrence sequence generation produces 16,594 augmented sequences. Word2Vec training completes in under 30 seconds on a single CPU core. Agglomerative clustering over 6,221 documents takes approximately 2 minutes. The entire offline pipeline runs in under 20 minutes, making it practical for periodic KB reorganization.

At inference time, the hybrid retrieval adds negligible latency: cluster lookup is  $O(1)$  via a pre-computed dictionary, and the expansion step adds at most  $k_v \times |\bar{C}|$  candidates (approximately 15) to the re-ranking pool. The total retrieval overhead is dominated by the original embedding similarity computation, not the expansion.

## 5 Experimental Setup

### 5.1 Datasets

**WixQA.** Our primary evaluation uses WixQA (Cohen et al., 2025), an enterprise

RAG benchmark containing 6,221 customer support articles from the Wix Help Center knowledge base. The dataset includes 400 expert-labeled queries: 200 expert-written queries with multi-step answers and 200 simulated queries validated by domain experts. Each query is annotated with ground-truth article IDs from the KB, enabling evaluation with real relevance labels.

**E-Commerce Support.** For cross-domain validation, we evaluate on an e-commerce customer support corpus containing 1,000 conversations across 10 product categories and multiple issue types. Co-occurrence sessions are defined by shared issue-category and product-category metadata, providing naturally defined user sessions without synthetic construction.

## 5.2 Session Construction

Sessions correspond to co-occurrence clusters, where each cluster represents a group of articles that collectively serve a user’s information-seeking episode. We construct 200 evaluation sessions from the WixQA clusters, with a mean size of 7.9 documents (range: 3 to 70). For each evaluation trial, we randomly sample one entry query from the session and measure retrieval coverage over the full session.

## 5.3 Baselines

We compare our hybrid retrieval against two baselines: (1) **Vanilla RAG**, which retrieves the top- $k$  documents by cosine similarity between query and document embeddings; and (2) **Cross-encoder re-ranking**, which retrieves the top-30 by embedding similarity and then re-ranks using ms-marco-MiniLM-L-6-v2 cross-encoder.

## 5.4 Embedding Models

To demonstrate encoder-agnostic gains, we evaluate across four embedding models spanning different architectures, training objectives, and model sizes: all-MiniLM-L6-v2 (22M parameters, general-purpose), bge-base-en-v1.5 (109M, retrieval-optimized), gte-base (109M, text clustering), and e5-base-v2 (109M, weakly supervised).

## 5.5 Statistical Methodology

All reported confidence intervals are computed via bootstrap resampling with 1,000 samples at the 95% level. We seed all random operations (session entry point selection, sequence generation) with a fixed seed for reproducibility.

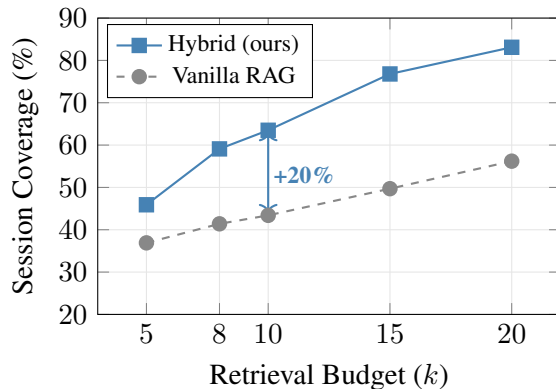


Figure 3: Session coverage vs. retrieval budget. The shaded area represents the coverage gap that cluster expansion closes. Notably, the gap *widens* with  $k$ .

$k$	Vanilla	Hybrid	$\Delta$	Rel. $\Delta$
5	36.9%	45.9%	+9.0%	+24%
8	41.4%	<b>59.1%</b>	+17.7%	+43%
10	43.4%	63.5%	+20.1%	+46%
15	49.7%	76.8%	+27.1%	+55%
20	56.2%	83.1%	+26.8%	+48%

Table 1: Session coverage at varying retrieval budgets. Bold indicates the primary operating point ( $k=8$ ).

# 6 Results

## 6.1 Session Coverage vs. Retrieval Budget

Table 1 and Figure 3 report session coverage as the retrieval budget varies. The coverage gap between vanilla and hybrid *widens* with  $k$ : from +9.0% at  $k=5$  to +27.1% at  $k=15$ . This scaling property is important because it means that as context windows grow and retrieval budgets increase, the relative value of cluster expansion grows rather than saturates. At  $k=20$ , hybrid retrieval achieves 83.1% session coverage, compared to 56.2% for vanilla.

## 6.2 Retrieval Efficiency

Table 2 and Figure 4 quantify the practical efficiency gain. To reach 70% session coverage, vanilla RAG requires 4.0 calls on average while hybrid retrieval requires only 2.6, a 34% reduction. At the 80% threshold, the savings increase to 1.7 fewer calls per session. In production systems that handle millions of sessions, this translates directly to reduced latency, lower API costs, and fewer user interactions needed to resolve each support episode.

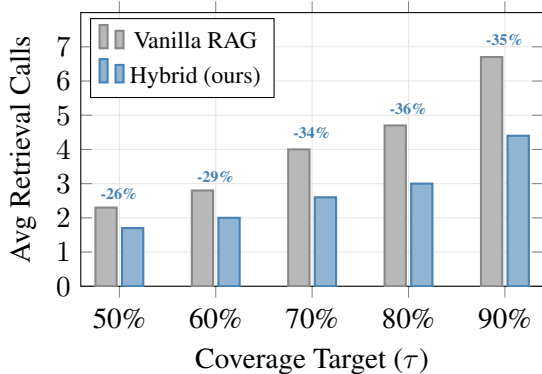


Figure 4: Retrieval calls to reach session coverage thresholds. Hybrid retrieval requires 26 to 36% fewer calls across all targets.

$\tau$	Vanilla	Hybrid	Reduction	Saved
50%	2.3	1.7	26%	0.6
60%	2.8	2.0	29%	0.8
70%	4.0	<b>2.6</b>	<b>34%</b>	1.4
80%	4.7	3.0	36%	1.7
90%	6.7	4.4	35%	2.4

Table 2: Retrieval calls needed to reach coverage thresholds  $\tau$ .

### 6.3 Effect of Session Complexity

Table 3 stratifies results by the number of documents in the session. The method provides substantial gains at every complexity level, with all 95% bootstrap confidence intervals excluding zero. The largest absolute gain appears on complex sessions (8 to 15 documents): +17.6% with a CI lower bound of +11.8%. For simple sessions (3 to 4 documents), coverage jumps from 52.0% to 70.8%. The diminishing absolute gain on very complex sessions (16+ documents) is expected: a retrieval budget of  $k=8$  cannot cover 16 or more documents in a single call regardless of the retrieval strategy.

### 6.4 Encoder Robustness

Table 4 demonstrates that the cluster structure, which is learned once from co-occurrence patterns, improves retrieval across all four embedding models. The deltas range from +17.1% to +20.6% absolute, with every bootstrap CI excluding zero. This confirms that co-occurrence captures information that is orthogonal to any specific embedding model’s representation, and that the offline cluster structure is a genuinely encoder-agnostic resource.

Type	$N$	Van.	Hyb.	$\Delta$	95% CI
Simple (3–4)	92	52.0	<b>70.8</b>	+18.8	[13.1, 24.3]
Medium (5–7)	48	41.5	58.1	+16.7	[10.7, 22.9]
Complex (8–15)	41	28.3	45.9	+17.6	[11.8, 23.1]
V. Cmplx (16+)	19	15.2	23.8	+8.6	[5.5, 11.9]

Table 3: Session coverage (%) by complexity with 95% bootstrap CIs. All intervals exclude zero.

Encoder	Van.	Hyb.	$\Delta$	95% CI
MiniLM-L6-v2	41.1	58.2	+17.1	[14.0, 20.2]
bge-base-v1.5	39.8	57.1	+17.2	[14.2, 20.7]
gte-base	40.4	60.8	+20.4	[17.0, 23.8]
e5-base-v2	39.8	<b>60.4</b>	+20.6	[17.1, 23.9]

Table 4: Session coverage (%) across encoders ( $k=8$ ) with 95% bootstrap CIs. All intervals exclude zero. Clusters are learned once and benefit all encoders.

## 6.5 Domain Generalization

We apply Latent Dirichlet Allocation to identify six functional domains within WixQA (Table 5). The hybrid method improves coverage in **all six domains**, with gains ranging from +12.7% (Apps & Email) to +21.5% (Editor & Studio). This consistency across topically diverse domains confirms that the method is not specific to any particular content type or user behavior pattern within the KB.

## 6.6 Cross-Dataset Validation

To validate beyond a single benchmark, we evaluate on an e-commerce customer support corpus with 1,000 documents and 94 naturally defined sessions. Session coverage improves from 32.2% to 43.3% (+11.1% absolute, +34% relative), confirming that the method generalizes to a different domain, document structure, and session definition.

## 7 Analysis

### 7.1 Semantic Similarity vs. Co-occurrence

We compute the Pearson correlation between semantic similarity (cosine in MiniLM embedding space) and co-occurrence strength for 10,000 sampled document pairs (Figure 5). The correlation is weak ( $r \approx 0.2$ ), confirming that documents users need together are *not* necessarily the documents that look alike in embedding space. This weak correlation is precisely what makes co-occurrence clustering valuable: it provides a complementary signal that semantic embeddings structurally cannot capture, and cluster expansion surfaces the “low

Domain	$N$	Van.	Hyb.	$\Delta$
Bookings & Payments	37	34.2%	50.8%	+16.6%
Apps & Email	28	40.0%	52.6%	+12.7%
Editor & Studio	58	46.3%	<b>67.8%</b>	+21.5%
Plans & Pricing	13	52.8%	71.1%	+18.3%
Media & Domains	26	50.0%	63.0%	+13.0%
Stores & Blog/CMS	38	37.6%	51.9%	+14.3%

Table 5: Session coverage across six WixQA functional domains. Gains are positive in all six domains.

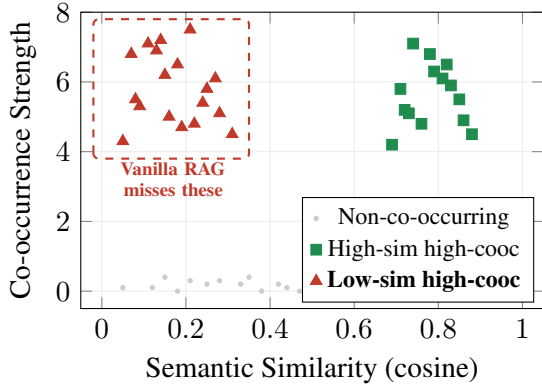


Figure 5: Semantic similarity vs. co-occurrence strength (schematic from 10K sampled pairs). The **red triangle** region (low similarity, high co-occurrence) contains document pairs that vanilla RAG cannot surface together. Weak Pearson correlation ( $r=0.20$ ) confirms the signals are complementary.

similarity, high co-occurrence” document pairs that vanilla RAG systematically misses.

## 7.2 Cluster Quality Analysis

Table 6 reports detailed clustering statistics. The 1,244 clusters have a mean size of 5.0 documents with a long-tailed distribution: most clusters contain 3 to 5 documents (corresponding to simple user sessions), while a small number of large clusters (up to 70 documents) capture broad topics like “account management” that touch many articles.

The silhouette score of 0.205 indicates moderate cluster cohesion. This moderate (rather than high) score is expected and actually desirable: co-occurrence clusters intentionally group semantically *diverse* documents, so high cohesion in the semantic embedding space would indicate that the clusters are simply recapitulating semantic similarity rather than adding new information.

The within-cluster co-occurrence rate of 25.9% means that approximately one in four co-occurring document pairs land in the same cluster. This is substantially above the random baseline of  $1/M \approx 0.08\%$ , confirming meaningful structure. The rate

Statistic	Value
Original KB size	6,221 documents
Number of clusters $M$	1,244
Compression ratio	20.0%
Mean cluster size	5.0
Median cluster size	4
Max cluster size	70
Silhouette score	0.205
Within-cluster co-occ rate	25.9%
Word2Vec vocabulary coverage	98.8%
Augmented co-occ sequences	16,594

Table 6: Clustering statistics on WixQA.

#	Document Title
1	Connecting a Domain You Already Own
2	Transferring a Domain to Wix
3	Getting a Free Domain with Premium Plans
4	Changing Your Site Template
5	Customizing Your Site’s Favicon

Table 7: Example cluster containing semantically diverse but functionally related documents. Documents 1–3 cover domain management; documents 4–5 cover site design. Users setting up a new website commonly need all five.

is not higher because the co-occurrence sequences draw from three signal sources with different properties, and the clustering finds a compromise that respects all three.

## 7.3 Qualitative Example

To illustrate the method’s behavior concretely, consider a representative cluster:

Documents 1 through 3 form a tight semantic group (domain management), while documents 4 and 5 are semantically distant (site design). Yet all five are commonly needed by users setting up a new website. When a user asks about connecting their domain, vanilla RAG retrieves documents 1 through 3 but misses 4 and 5 entirely. Cluster expansion adds documents 4 and 5 to the candidate pool, and the re-ranker surfaces them if they match the query context.

## 7.4 Precision Trade-off

Cluster expansion trades marginal first-hit precision for substantially broader coverage. At  $k=8$ , Hits@K decreases slightly from 96.0% to 93.0%. This is the correct trade-off for session-oriented settings: covering 58% of the user’s full information need is more valuable than a 3% improvement in whether the single best document appears in the top result. In practice, this trade-off can be tuned

by adjusting  $k_v$ : increasing  $k_v$  from 3 to 5 preserves more vanilla precision at the cost of fewer expansion slots.

### 7.5 Boundary Condition: When the Method Does Not Help

On HotpotQA (Yang et al., 2018), where each question has an isolated paragraph set with no cross-query document reuse, the method shows no gain ( $\Delta = -1.4\%$ ). This negative result is informative: it confirms that the improvement requires a *persistent* knowledge base where documents are reused across multiple user sessions. In such settings, co-occurrence patterns form naturally from repeated access. In benchmarks where each question constructs an ad-hoc document set, no co-occurrence signal can emerge. This boundary condition confirms that gains are driven by learned co-occurrence patterns rather than by an artifact of the expansion mechanism itself.

## 8 Conclusion

We propose that RAG evaluation should shift from single-query recall to session-level coverage, and demonstrate a simple method to close the resulting coverage gap: reorganize the KB offline using co-occurrence-aware clustering, then expand retrieval candidates through cluster neighborhoods at query time. A single retrieval call over a reorganized KB covers 58% of a user’s session-level information need (compared to 41% for standard RAG) while requiring 34% fewer calls to reach practical coverage thresholds. The method is encoder-agnostic (+17 to 21% across four models), domain-general (gains in 6 out of 6 domains), and provides 80% KB compression as a side benefit.

For future work, we plan to incorporate real user navigation logs to replace synthetic co-occurrence sequences and measure how the quality of the co-occurrence signal affects downstream coverage gains. We also plan to evaluate the impact on end-to-end generation quality using LLM-as-judge metrics, as higher retrieval coverage does not automatically translate to better generated answers if the LLM cannot effectively use the broader context. Finally, we intend to explore adaptive cluster expansion, where the number of expanded candidates scales with query ambiguity, and to investigate combining our pre-retrieval KB reorganization with post-retrieval compression methods like EDC<sup>2</sup>-RAG (Li et al., 2025) for end-to-end session-

level RAG optimization.

## Limitations

Our co-occurrence sequences are constructed from embedding neighborhoods and synthetic QA pairs rather than real user interaction logs. While the ground-truth article groups from WixQA provide a real co-occurrence signal and the HotpotQA boundary experiment confirms that persistent document reuse is required, real enterprise usage data would likely produce stronger co-occurrence patterns and larger gains. We were unable to use real navigation logs due to the absence of publicly available enterprise session data, a common constraint in this domain.

Session-level evaluation uses cluster-derived sessions as a proxy for real user sessions; evaluation on logged multi-turn conversations would strengthen the claims.

We evaluate retrieval coverage but not downstream generation quality. Higher coverage introduces more diverse context into the LLM’s input, which may trigger the “lost in the middle” phenomenon (Liu et al., 2024) where models underutilize information in the middle of long contexts. Whether the 58% session coverage translates to proportionally better generated answers remains an open question that we leave to future work.

The 40% random jump probability in embedding walks was set heuristically; a systematic search over this hyperparameter could improve results. Finally, while we evaluate across four encoders, six domains, and two datasets, additional enterprise KB evaluations with real user session logs would further establish generalizability.

## Ethics Statement

This work uses publicly available datasets (WixQA under MIT license) and does not involve human subjects, private user data, or personally identifiable information. The synthetic QA pairs are generated from public knowledge base articles. Our method reorganizes existing knowledge bases and does not generate new content, so it does not introduce additional hallucination risk beyond what is inherent in the underlying RAG system. We note that co-occurrence patterns learned from real user logs, if used in future work, would require appropriate anonymization and privacy protections.

## References

- Simon Akesson and Frances A. Santos. 2024. [Clustered retrieved augmented generation \(CRAG\)](#). *Preprint*, arXiv:2406.00029.
- Oren Barkan and Noam Koenigstein. 2017. [Item2vec: Neural item embedding for collaborative filtering](#). *Preprint*, arXiv:1603.04259.
- Dvir Cohen, Lin Burg, Sviatoslav Pykhnivskiy, Hagit Gur, Stanislav Kovynov, Olga Atzmon, and Gilad Barkan. 2025. [WixQA: A multi-dataset benchmark for enterprise retrieval-augmented generation](#). *Preprint*, arXiv:2505.08643.
- Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. [RAGAS: Automated evaluation of retrieval augmented generation](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.
- Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. [Precise zero-shot dense retrieval without relevance labels](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.
- 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.
- Weitao Li, Xiangyu Zhang, Kaiming Liu, Xuanyu Lei, Weizhi Ma, and Yang Liu. 2025. [Efficient dynamic clustering-based document compression for retrieval-augmented-generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9833–9849, Suzhou, China. Association for Computational Linguistics.
- 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.
- Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. [Fine-tuning LLaMA for multi-stage text retrieval](#). *Preprint*, arXiv:2310.08319.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient estimation of word representations in vector space](#). *Preprint*, arXiv:1301.3781.
- Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. [RAPTOR: Recursive abstractive processing for tree-organized retrieval](#). *Preprint*, arXiv:2401.18059.
- Orion Weller, Michael Boratko, Iftexhar Naim, and Jinhyuk Lee. 2026. [On the theoretical limitations of embedding-based retrieval](#). *Preprint*, arXiv:2508.21038.
- Yifan Xu, Vipul Gupta, Rohit Aggarwal, Varsha Mahadevan, and Bhaskar Krishnamachari. 2025. [Cluster-based adaptive retrieval: Dynamic context selection for RAG applications](#). *Preprint*, arXiv:2511.14769.
- 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.

## A Single-Query Retrieval Results

For completeness, we report standard single-query retrieval metrics on the 400 WixQA expert queries (Table 8). We evaluate a boost-based scoring variant which adds a cluster-relevance signal to direct document similarity with weight  $\alpha=0.2$ . This provides modest but consistent gains over vanilla retrieval on single-query metrics. The gains are small because WixQA queries are predominantly single-aspect; the method’s primary value is at the session level (Section 6).

Method	W. Recall	Hits@K	MRR
Vanilla	52.8%	57.5%	0.359
Reranker	53.1%	58.0%	0.334
Boost ( $\alpha=0.2$ )	55.2%	59.8%	0.363

Table 8: Single-query metrics on 400 WixQA expert queries.

## B Hyperparameter Sensitivity

**Vanilla slots  $k_v$ .** Table 9 shows session coverage as  $k_v$  varies with  $k=8$  fixed. Lower  $k_v$  allocates more slots to cluster expansion, increasing coverage at the cost of first-hit precision. We select  $k_v=3$  as it provides the best balance.

**Boost weight  $\alpha$ .** For the boost-based variant used in single-query evaluation, Table 10 shows the effect of  $\alpha$ . The value  $\alpha=0.2$  provides the best weighted recall; higher values degrade performance as the cluster signal overwhelms direct similarity.

$k_v$	Coverage	Hits@K	$\Delta$ Cov
1	61.2%	87.5%	+20.1%
2	59.8%	90.0%	+18.7%
3	58.2%	93.0%	+17.1%
5	50.2%	95.5%	+9.1%
7	43.5%	96.0%	+2.4%

Table 9: Effect of vanilla slots  $k_v$  on session coverage and Hits@K ( $k=8$ ).  $k_v=3$  balances coverage with precision.

$\alpha$	W. Recall	Hits@K	$\Delta$ WR
0.1	54.1%	58.8%	+1.2%
0.2	<b>55.2%</b>	59.8%	+2.4%
0.3	54.2%	58.8%	+1.3%
0.4	52.2%	56.5%	-0.7%
0.5	50.8%	55.0%	-2.0%

Table 10: Effect of boost weight  $\alpha$  on single-query metrics.

## C Cross-Dataset Summary

Table 11 summarizes results across all evaluation settings, including the HotpotQA boundary experiment.

Dataset	Domain	Docs	Van.	Hyb.	$\Delta$
WixQA	Web Support	6,221	41%	58%	+17%
E-Commerce	E-Commerce	1,000	32%	43%	+11%
HotpotQA	Open-domain	5,842	35.8%	34.4%	-1.4%

Table 11: Cross-dataset session coverage ( $k=8$ ). Gains are consistent on persistent KBs and absent on ad-hoc paragraph sets (HotpotQA), confirming the method requires cross-session document reuse.

## D Reproducibility Details

The offline pipeline (QA generation, sequence construction, Word2Vec training, and clustering) runs in under 20 minutes on a single CPU core, excluding API time for QA generation (approximately 6 minutes with 30 concurrent threads). All embedding models are loaded via the sentence-transformers library. WixQA is available under MIT license at Wix/WixQA on HuggingFace.

Parameter	Value
Word2Vec architecture	CBOW
Embedding dimension	100
Window size	2
Negative samples	10
Training epochs	30
Random jump probability	0.40
Clustering method	Agglomerative (cosine, avg)
Target compression	20% ( $M = \lceil N/5 \rceil$ )
Vanilla slots $k_v$	3
Retrieval budget $k$	8
QA generation model	GPT-4.1-nano
QA pairs per document	10
Bootstrap samples	1,000
Random seed	42

Table 12: Full hyperparameter configuration.

# Author Index

- Banar, Nikolay, 34  
Bugaud, Zacharie, 13
- Cheng, Cecilia, 173
- Daelemans, Walter, 34  
Deshpande, Pranamya Nilesh, 165  
Dhayalkar, Sahil Rajesh, 78  
Du, Yixuan, 115
- Hao, Joanna, 25  
Hu, Xiyang, 115
- Jiang, Shanduoqiao, 25
- Kamel, Adam, 138  
Kliocaite, Marija, 34  
Konavoor, Aiswarya, 165
- Li, Yichen, 105  
Liu, Shiyan, 105  
Lotfi, Ehsan, 34
- Nakka, Sai Asish, 25
- Panat, Sreedath, 165  
Polyzou, Agoritsa, 60
- Ranjan, Ravi, 60  
Ratnakar, Shivam, 173
- Song, Zihe, 90
- Teillers, Wout, 147
- Valdenegro-Toro, Matias, 147  
Van Nooten, Jens, 34  
Vijayakumar, Chaya, 173
- Wang, Jindong, 90  
Wang, Ziyi, 115
- Xu, Eric, 138  
Xu, Haoyan, 115  
Xu, Ke, 90
- Yang, Liangwei, 90  
Yu, Chenxiao, 115  
Yu, Philip S., 90  
Yu, Xinqiang, 129
- Zaghouani, Wajdi, 1  
Zhang, Weizhi, 90  
Zhao, Jianing, 129  
Zhao, Yue, 115  
Zhi, Xiaoquan, 129  
Zhu, Yixuan, 173  
Zhu, Yuanjie, 90