Adaptation of Large Language Models

Zixuan Ke **Yifei Ming** **Shafiq Joty**
Salesforce AI Research
{zixuan.ke,yifei.ming,sjoty}@salesforce.com

1 Brief Description

This tutorial on adaptation of Large Language Models (LLMs) is designed to address the growing demand for models that go beyond the static capabilities of generic LLMs by providing an overview of dynamic, domain-specific, and task-adaptive LLM adaptation techniques. While general LLMs have demonstrated strong generalization across a variety of tasks, they often struggle to perform well in specialized domains such as finance, healthcare, and code generation for underrepresented languages. Additionally, their static nature limits their ability to evolve with the changing world, and they are often extremely large in size, making them impractical and costly to deploy at scale. As a result, the adaptation of LLMs has drawn much attention since the birth of LLMs and is of core importance, both for industry, which focuses on serving its targeted users, and academia, which can greatly benefit from small but powerful LLMs.

To our knowledge, there has been no similar tutorial on this topic in the past few years[1]. To address this gap, this tutorial aims to provide an overview of the LLM adaptation techniques. We start with an introduction to LLM adaptation, from both the **data perspective**, which has been widely accepted as one of the most important ingredients in LLM training, and the **model perspective**, which focuses on the training strategies to adapt the LLMs.

In Session 2, we emphasize how the **evaluation metrics and benchmarks** are different from other techniques. There is a growing recognition that evaluating adapted LLMs should not focus solely on their performance within the target domains or tasks. While this remains important, an emerging consensus is that adaptation should not be at the cost of sacrificing what it has already learned. An ideal evaluation protocol should therefore assess not only specialized task performance but also the preservation of general knowledge. Furthermore, as we will present in this tutorial, adapting an LLM usually involves multiple training stages, where each stage contributes to the final outcome. This multi-stage process requires more comprehensive evaluation methods, as conventional metrics may not capture the cumulative impact of these stages. As such, researchers are increasingly focused on designing evaluation metrics and benchmarks that assess both the specialized knowledge gained and the general knowledge retained.

Slot	Session
Session 1: Introduction and Motivation	
14:00 - 14:05	Tutorial presenters introduction
14:05 - 14:10	Challenges with general-purpose LLMs
14:10 - 14:20	Overview and use cases of LLM adaptation
14:20 - 14:30	Key concepts
14:30 - 14:40	Research questions in data and model
Session 2: Evaluation and benchmark	
14:40 - 14:50	General and specialized knowledge
14:50 - 15:00	Evaluation metrics
Session 3: Parametric Knowledge Adaptation	
15:00 - 15:15	Domain-adaptive Pre-training
15:15 - 15:30	Instruction Tuning
15:30 - 16:00	Coffee Break
16:00 - 16:15	Preference Learning
16:15 - 16:30	Model Editing
Session 4: Semi-Parametric Knowledge Adaptation	
16:30 - 16:45	Retrieval-augmented Generation
16:45 - 17:00	Agent-based Integration
17:00 - 17:30	Summary, Discussion & QA

Table 1: Example tutorial schedule.

After establishing the problems in the aforementioned sessions, we explore various **adaptation techniques**. When powerful general LLMs first emerged, many believed that prompt engineering alone, without any actual model training (i.e., updating the model's parameters), was sufficient to adapt them for new tasks. However, recently we have seen more researchers begin finetuning model wights. This shift is understandable not only because training typically leads to better results on specialized tasks, but also because training has become more affordable due to the growing number

[1] https://www.aclweb.org/adminwiki/index.php/Past_tutorials

Proceedings of the 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorial Abstracts, pages 30–37
May 1, 2025 ©2025 Association for Computational Linguistics

of techniques that enable efficient training. For example, the training cost increased from $900 for the original Transformer to over $4 million to train the GPT-3 (davinci), but it has recently dropped to around $0.8 million to train a GPT-3 on-par model like Phi3.5 (Maslej et al., 2024). In light of this trend, this tutorial will focus on increasingly popular methods that adapt LLMs through training.

We categorize adaptation techniques into two main families. The first is **parametric knowledge adaptation**, which focuses on updating the parametric knowledge within LLMs, including methods like Domain-Adaptive Pre-Training (DAPT), Instructional Tuning (IT), and Preference Learning (PL) via human or model feedback. Additionally, we will discuss real-time adaptation techniques, including model editing, which allows LLMs to be updated dynamically in production environments. The second kind of adaptation is **semi-parametric knowledge adaptation**, where the goal is to update LLM parameters to better leverage external knowledge or tools (e.g., documents or functions) through techniques like retrieval-augmented generation (RAG) and agent-based systems.

2 Detailed Outline

This will be a **three-hour** tutorial devoted to the **cutting-edge topic** of the adaptation of LLMs. Table 1 gives an overview.

1. **Introduction and Motivation**

1.1. **Challenges with General-purpose LLMs.** While LLMs have demonstrated strong generalization capabilities, they are general and static. They often fall short in highly specialized domains or tasks. These LLMs struggle to adapt to evolving requirements and exhibit high computational overhead due to their fixed nature. This section will highlight key limitations of them, with detailed examples and statistics to illustrate why adaptation is necessary for handling domain-specific or task-specific demands.

1.2. **Overview and Use Cases of LLM Adaptation.** LLM Adaptation adapts a pre-trained LLM to specialized use cases. It offers practical solutions for domain-specific applications, such as in finance and healthcare, as well as task-specific applications like retrieval-augmented generation and code generation. Personalized systems also benefit from adaptive models, which can fine-tune responses based on individual user needs. This section will showcase real-world use cases that demonstrate the critical need for LLM adaptability to address specialized scenarios.

1.3. **Key Concepts.** Before exploring detailed methodologies, we will introduce foundational concepts essential to understanding adaptive LLMs. These include distinctions between domain vs. task adaptation, continual learning vs. LLM adaptation, Parameter-Efficient Fine-Tuning (PEFT), different stages of post-training (e.g., Domain-Adaptive Pre-Training and Supervised Fine-Tuning), long-context LLMs, retrieval-augmented generation (RAG), agentic approach. These concepts will set the stage for the following technical sections.

1.4. **Research Questions in Data and Model Perspective.** In this tutorial, we aim to address several critical research questions. In data perspective, we focus on what gives a good data mixture and how to obtain high-quality data. In model perspective, we ask what constitutes an effective training recipe for adaptive LLMs, how can we properly evaluate adaptive models across different domains and tasks, what strategies can be used to prevent data leakage during post-training, and how can we scale up post-training efficiently. These questions will guide the subsequent discussions on evaluation and training.

2. **Evaluation and Benchmarks.** Evaluating adaptive LLMs involves a multidimensional approach. We will cover methods for assessing generalization, domain- and task-specific performance, and real-time effectiveness.

2.1. **General and Specialized Knowledge.** When adapting a model to specialize in specific knowledge, it is critical not to lose already learned general knowledge. A robust evaluation protocol should account for both. We will introduce methods to evaluate both types.

2.2. **Evaluation Metrics** . Numerous metrics have been discussed in the literature, such as perplexity, needle-in-haystack evaluations, and downstream task performance. We will compare and analyze the effectiveness of these metrics in evaluating adaptive LLMs.

3. **Parametric Knowledge Adaptation.** Multiple stages are undertaken to refine and specialize their knowledge for specific tasks and domains.

3.1. **Domain-Adaptive Pre-Training (DAPT).** Discuss the data and objective of DAPT that learn domain-specific background knowledge using raw text with next token prediction.

3.2. **Instruction Tuning (IT).** Discuss the data and objective of IT, also known as Supervised Fine-Tuning (SFT). It can be full model fine-tuning or parameter-efficient fine-tuning, with focus on instruction following ability.

3.3. **Preference Learning (PL)** Discuss the PL, where extra supervision on negative cases and delayed reward are introduced in this stage. The preference feedback can come from humans, AI models, or programmatic metrics. It can be done in online or offline (direct preference optimization; DPO) manner.

3.4. **Model Editing.** Discuss the techniques that allow for targeted updates in real-time, improving adaptability in production environments without the need for expensive retraining.

4. **Semi-parametric Knowledge Integration**

4.1. **Retrieval-augmented Generation (RAG)** How RAG allows models to dynamically retrieve external knowledge, and significantly enhance performance on specialized tasks or when up-to-date information is needed.

4.2. **Agent-based Integration.** This focuses more on the agentic aspect, highlighting the model as an agent that performs multi-step reasoning and interacts with external APIs and dynamic environments. It draws attention to the system's autonomous decision-making capacity and its ability to operate as an agent in complex environments.

5. **Summary, Discussion & QA**

3 Proportion of Other Researchers' Work, Reading list and Prerequisite

LLM adaptation involved almost all domains and tasks, making it become the main focus of industry and academic research. Our tutorial is designed to provide a comprehensive overview of this broad and rapidly evolving field, ensuring that the content reflects a wide range of techniques and contributions. As a result, the tutorial naturally includes a substantial amount of work from other researchers.

The prerequisite includes familiarity with basic knowledge of machine learning, NLP and LLM. Knowledge of continual learning is a plus. Here are a few papers that lay a foundation for this area and we encourage attendees to read the papers before the tutorial to familiarize themselves with foundational concepts and approaches:

- Continual pre-training of language models (Ke et al., 2023)
- Locating and editing factual associations in GPT (Meng et al., 2023a)
- Retrieval augmented language model pre-training (Guu et al., 2020)

Below, we provide a detailed reference to the works we plan to cover. While we will give pointers to dozens of relevant papers over the course of the tutorial, we plan to cover around 7-8 research papers in close detail. Only 1-2 of the "deep dive" papers will come from the presenter team.

Continual Learning learns a sequence of tasks sequentially without forgetting (Chen and Liu, 2018; McCloskey and Cohen, 1989; Van de Ven and Tolias, 2019; Mai et al., 2022; Aljundi et al., 2019; Ke and Liu, 2022), which is closely related to LLM adaptation. Typical approaches include *regularization-based methods* that regularize parameter updates to preserve important parameters (Kirkpatrick et al., 2016; Seff et al., 2017); *modular-based methods* that dynamically modify the architecture (Serrà et al., 2018; Wortsman et al., 2020); and *replay-based method* that recall previous experiences (Rebuffi et al., 2017; Wang et al., 2020).

Parametric Knowledge Adaptation adapts LLMs by updating its parametric knowledge. It has been widely adopted across board domains, such as code (Nijkamp et al., 2022), medical (Luo et al., 2023), law (Colombo et al., 2024), mathematics (Azerbayev et al., 2023), multi-lingual (Chen et al., 2024a) and finance (Ke et al., 2025; Xie et al., 2023a; Writer, 2024) and tasks such as function calling (Zhang et al., 2024b). Some focus on domain-specific or task-specific data curation (Yang et al., 2024b), mixture ratio (Que et al., 2024), data-efficiency (Xie et al., 2023b), hyper-parameters (Parmar et al., 2024) or training recipe (Jiang et al., 2024; Yu et al., 2024; Gao et al., 2024).

Semi-Parametric Knowledge Adaptation augments LLMs with relevant information retrieved from various knowledge sources, i.e., RAG or external functions, i.e., agent-based intergation.

Retrieval-Augmented Generation (RAG) is proven effective across numerous NLP tasks, including language modeling (Borgeaud et al., 2022; Khandelwal et al., 2020; Shi et al., 2023), question answering (Ke et al., 2024; Lewis et al., 2020; Izacard et al., 2022; de Jong et al., 2023; De Jong et al., 2023; Shi et al., 2023; Guu et al., 2020; Izacard and Grave, 2020; Xu et al., 2023), fact versification (Lewis et al., 2020) and text generation (Lewis et al., 2020). Specifically, RAG utilizes input as a query and comprises two main components: a *retriever* retrieves a set of items from a side corpus, and a *LLM* incorporates the retrieved items, as additional information in the input context, and makes final predictions. Depending on which components are subject to updates, typical approaches can be categorized into three families: *adapting retrievers and LLMs jointly*, which is the most widely used setting of RAG (Izacard et al., 2022; Khandelwal et al., 2020; Wu et al., 2022; Guu et al., 2020; Lewis et al., 2020); *adapting LLMs only*, which avoid the costly updates of retrievers and document index (Izacard and Grave, 2020; De Jong et al., 2023; de Jong et al., 2023); and *adapting retrievers only*, which compatible with black-box LLMs (Shi et al., 2023; Xu et al., 2023).

Model Editing aims to adapt LLM in real-time without the need for expensive training or even without training (Zhang et al., 2024c; Wang et al., 2024). Typical approaches include *locate-and-edit* that first locate the position of the knowledge in LLMs and edit the KV cache (Meng et al., 2023a,b; Yang et al., 2024a; Gupta et al., 2024a,b), *meta-learning* that employ hyper-network to learn the editing (Tan et al., 2024; Cao et al., 2021; Mitchell et al., 2022) and *memory-based* that use memory elements to store and manipulate information during editing (Zhu et al., 2020; Ni et al., 2024; Zheng et al., 2023). As of now, editing is still in its infancy and suffers from serious forgetting, in particular when editing a sequence of samples.

Agent-based Integration augments the LLMs with external functions. Typical approaches include *fine-tuning the LLMs* for better function calling capacities (Qin et al., 2023; Chen et al., 2023; Patil et al., 2023; Zeng et al., 2023; Zhang et al., 2024a; Yin et al., 2024; Chen et al., 2024b; Li et al., 2023; Tang et al., 2023); and *prompting techniques* (Wei et al., 2023; Yao et al., 2022; Shinn et al., 2023)

4 Target Audience and Estimated Size

Researchers, graduate students, and practitioners who are interested in NLP and machine learning. The tutorial will is particularly beneficial to people who intend to develop machine learning and NLP techniques and applications that can adapt the LLM to specialized domains or tasks while preserving the original learned knowledge in the LLM. This tutorial has not been given before. Due to the increasing popularity of LLM and its adpatation, we estimate that there will be 100 to 150 people attending the tutorial.

5 Diversity Consideration

The topic of LLM adaptation will be inclusive for a broad range of communities. It will be of interest to industrial practitioners as they aim to deploy the best adaptive models for their targeted customers. Additionally, academic researchers will find value in this topic, as it essentially addresses a continual learning problem, enabling LLMs to adapt to new environments without forgetting—an essential step toward advancing Artificial General Intelligence. This topic also makes it easier for users to engage with LLMs, as it involves techniques to train smaller models that can be deployed efficiently. It naturally spans a wide range of domains and tasks, making it relevant to participants interested in various specific areas.

Our presenters include both junior and senior researchers from industry and academia, and we also bring geographical diversity, with instructors from both North America and Asia. This diverse group of instructors will help attract a wide audience and encourage participation from various backgrounds.

We will make our tutorial materials digitally accessible to all participants. All materials will be publicly available before the tutorial. We will provide slides and discussion questions before the tutorial to the audience. During the tutorial sessions, we will work with student volunteers to encourage open dialogue and promote active listening, allowing participants to share their thoughts and experiences without fear of judgment. After the tutorial, we will actively collect feedback to identify areas for improvement related to diversity and inclusion and share it with future tutorial presenters. Our presenter team will share our tutorial with a worldwide audience by promoting it on social media, and to diverse research communities. We will work with ACL/NAACL/EMNLP D&I teams, and consult resources such as the BIG directory to diversify our audience participation.

6 Ethics Statement

The tutorial will present general problems and algorithms related to LLM adaptation in NLP. All evaluation datasets are public domain benchmarks. Thus, the tutorial does not have any ethical issues.

7 Presenter Biography

Zixuan Ke is a research scientist at Salesforce AI Research, where he works with Dr. Shafiq Joty on continual pre-training of LLMs. Prior to that, he earned his Ph.D. at the University of Illinois, Chicago, where he worked with Dr. Bing Liu on continual learning. His research studies how to adapt the foundation models, particularly LLMs, for an ever-changing world characterized by emerging domains, events, topics, or information. He has published 25+ papers in top-tier conferences such as ICLR, ICML, NeurIPS, NAACL, ACL, and EMNLP with around 1.5K citations. He has served as an Area Chair and PC member for prestigious conferences such as ICLR, NeurIPS, ACL, EMNLP, and as a reviewer for leading journals such as Neural Networks, Neurocomputing, Artificial Intelligence, TKDE, and TPAMI, since 2021. He has given an invited short PhD course on continual learning at Aalborg University in the summer of 2022 and has also given several oral presentations at leading conferences. Further information about him, including past talks and slides, can be found at `https://vincent950129.github.io/`.

Yifei Ming is a Research Scientist at Salesforce AI Research. He received his Ph.D. in Computer Science from the University of Wisconsin-Madison in 2024. His research spans both empirical and theoretical studies, with a focus on understanding, benchmarking, and developing novel algorithms towards trustworthy foundation models (LLM and VLM). He has a prolific publication record with 20+ papers in top-tier AI conferences and journals. He co-organized several workshops and has served on the program committees of leading machine learning conferences and journals since 2021, including NeurIPS, ICLR, ICML, AAAI, EMNLP, IJCV, TPAMI, and NN. Further information can be found at `https://alvinmingsf.github.io/`.

Shafiq Joty is a research director at Salesforce Research and an Associate Professor at the Computer Science Department of the Nanyang Technological University. His research has contributed to over 30+ patents and 150+ papers in top-tier NLP and ML conferences and journals. He has served as a Program Chair of SIGDIAL'23, a member of the best paper award committees for ICLR'23 and NAACL'22, and in the capacity of a (senior) area chair for many leading NLP and ML conferences (e.g. NeurIPS, EMNLP, and ACL). He previously gave tutorials at EMNLP'23, IEEEVis'22, ACL'19, COLING'18 and ICDM'18. Further information about him, including past talks and slides, can be found at `https://raihanjoty.github.io/`.

References

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. 2019. Gradient based sample selection for online continual learning. In *NeurIPS*.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2023. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models.

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. 2023. Fireact: Toward language agent fine-tuning.

Jie Chen, Zhipeng Chen, Jiapeng Wang, Kun Zhou, Yutao Zhu, Jinhao Jiang, Yingqian Min, Wayne Xin Zhao, Zhicheng Dou, Jiaxin Mao, Yankai Lin, Ruihua Song, Jun Xu, Xu Chen, Rui Yan, Zhewei Wei, Di Hu, Wenbing Huang, and Ji-Rong Wen. 2024a. Towards effective and efficient continual pre-training of large language models.

Zehui Chen, Kuikun Liu, Qiuchen Wang, Wenwei Zhang, Jiangning Liu, Dahua Lin, Kai Chen, and Feng Zhao. 2024b. Agent-flan: Designing data and methods of effective agent tuning for large language models.

Zhiyuan Chen and Bing Liu. 2018. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3).

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre FT Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, et al. 2024. Saullm-7b: A pioneering large language model for law. *arXiv preprint arXiv:2403.03883*.

Michiel De Jong, Yury Zemlyanskiy, Nicholas FitzGerald, Joshua Ainslie, Sumit Sanghai, Fei Sha, and William W Cohen. 2023. Pre-computed memory or on-the-fly encoding? a hybrid approach to retrieval augmentation makes the most of your compute. In *International Conference on Machine Learning*, pages 7329–7342. PMLR.

Michiel de Jong, Yury Zemlyanskiy, Nicholas FitzGerald, Sumit Sanghai, William W Cohen, and Joshua Ainslie. 2023. Glimmer: generalized late-interaction memory reranker. *arXiv preprint arXiv:2306.10231*.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2024. How to train long-context language models (effectively).

Akshat Gupta, Sidharth Baskaran, and Gopala Anumanchipalli. 2024a. Rebuilding rome : Resolving model collapse during sequential model editing.

Akshat Gupta, Dev Sajnani, and Gopala Anumanchipalli. 2024b. A unified framework for model editing.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

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

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Zhengbao Jiang, Zhiqing Sun, Weijia Shi, Pedro Rodriguez, Chunting Zhou, Graham Neubig, Xi Victoria Lin, Wen-tau Yih, and Srinivasan Iyer. 2024. Instruction-tuned language models are better knowledge learners. *arXiv preprint arXiv:2402.12847*.

Zixuan Ke, Weize Kong, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Bridging the preference gap between retrievers and llms. *arXiv preprint arXiv:2401.06954*.

Zixuan Ke and Bing Liu. 2022. Continual learning of natural language processing tasks: A survey. In *Arkiv*.

Zixuan Ke, Yifei Ming, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. 2025. Demystifying domain-adaptive post-training for financial llms. *arXiv preprint arXiv:2501.04961*.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. *arXiv preprint arXiv:2302.03241*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *CoRR*.

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

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. API-bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Yushuai Wu, Mu Qiao, and Zaiqing Nie. 2023. Biomedgpt: Open multimodal generative pre-trained transformer for biomedicine. *arXiv preprint arXiv:2308.09442*.

Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. 2022. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469.

Nestor Maslej, Loredana Fattorini, Raymond Perrault, Vanessa Parli, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, and Jack Clark. 2024. The ai index 2024 annual report,.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023a. Locating and editing factual associations in gpt.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023b. Mass-editing memory in a transformer.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022. Fast model editing at scale.

Shiwen Ni, Dingwei Chen, Chengming Li, Xiping Hu, Ruifeng Xu, and Min Yang. 2024. Forgetting before learning: Utilizing parametric arithmetic for knowledge updating in large language models.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2022. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*.

Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Reuse, don't retrain: A recipe for continued pretraining of language models. *arXiv preprint arXiv:2407.07263*.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2023. Gorilla: Large language model connected with massive apis.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis.

Haoran Que, Jiaheng Liu, Ge Zhang, Chenchen Zhang, Xingwei Qu, Yinghao Ma, Feiyu Duan, Zhiqi Bai, Jiakai Wang, Yuanxing Zhang, Xu Tan, Jie Fu, Wenbo Su, Jiamang Wang, Lin Qu, and Bo Zheng. 2024. D-cpt law: Domain-specific continual pre-training scaling law for large language models.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. icarl: Incremental classifier and representation learning. In *CVPR*.

Ari Seff, Alex Beatson, Daniel Suo, and Han Liu. 2017. Continual learning in generative adversarial nets. *CoRR*, abs/1705.08395.

Joan Serrà, Didac Suris, Marius Miron, and Alexandros Karatzoglou. 2018. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*.

Chenmien Tan, Ge Zhang, and Jie Fu. 2024. Massive editing for large language models via meta learning.

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, Boxi Cao, and Le Sun. 2023. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases.

Gido M Van de Ven and Andreas S Tolias. 2019. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024. Knowledge editing for large language models: A survey.

Zirui Wang, Sanket Vaibhav Mehta, Barnabás Póczos, and Jaime Carbonell. 2020. Efficient meta lifelong-learning with limited memory. In *EMNLP*.

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

Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. 2020. Supermasks in superposition. In *NeurIPS*.

Writer. 2024. Palmyra-Fin-70B-32k: a powerful LLM designed for Finance. `https://dev.writer.com`.

Yuhuai Wu, Markus N Rabe, DeLesley Hutchins, and Christian Szegedy. 2022. Memorizing transformers. *arXiv preprint arXiv:2203.08913*.

Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023a. Pixiu: A large language model, instruction data and evaluation benchmark for finance.

Yong Xie, Karan Aggarwal, and Aitzaz Ahmad. 2023b. Efficient continual pre-training for building domain specific large language models.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*.

Wanli Yang, Fei Sun, Xinyu Ma, Xun Liu, Dawei Yin, and Xueqi Cheng. 2024a. The butterfly effect of model editing: Few edits can trigger large language models collapse.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candès, and Tatsunori Hashimoto. 2024b. Synthetic continued pretraining. *arXiv preprint arXiv:2409.07431*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. 2024. Agent lumos: Unified and modular training for open-source language agents.

Xiao Yu, Qingyang Wu, Yu Li, and Zhou Yu. 2024. Lions: An empirically optimized approach to align language models.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2023. Agenttuning: Enabling generalized agent abilities for llms.

Jianguo Zhang, Tian Lan, Rithesh Murthy, Zhiwei Liu, Weiran Yao, Juntao Tan, Thai Hoang, Liangwei Yang, Yihao Feng, Zuxin Liu, Tulika Awalgaonkar, Juan Carlos Niebles, Silvio Savarese, Shelby Heinecke, Huan Wang, and Caiming Xiong. 2024a. Agentohana: Design unified data and training pipeline for effective agent learning.

Jianguo Zhang, Tian Lan, Ming Zhu, Zuxin Liu, Thai Hoang, Shirley Kokane, Weiran Yao, Juntao Tan, Akshara Prabhakar, Haolin Chen, et al. 2024b. xlam: A family of large action models to empower ai agent systems. *arXiv preprint arXiv:2409.03215*.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. 2024c. A comprehensive study of knowledge editing for large language models.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning?

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models.