Overcoming Catastrophic Forgetting: A Novel Fine-Tuning Method

Anonymous ACL submission

Abstract

Despite remarkable advances in Large Language Models (LLMs), a persistent challenge remains: the potential for these models to acquire erroneous or outdated information from their training data. Direct fine-tuning with data containing new knowledge can be ineffective due to conflicts between old and new knowledge. This paper proposes a novel fine-tuning paradigm called Delicate Fine-Tuning (**DFT**) that leverages parametric arithmetic to pinpoint the location of knowledge and update only the minimal set of relevant parameters. Experimental results on two publicly available datasets demonstrate that our proposed DFT significantly improves the knowledge updating performance of full fine-tuning, consistently outperforming existing baselines in most cases.

1 Introduction

With the expanding applications of large language models (LLMs) across diverse domains (Achiam et al., 2023; Reid et al., 2024; Chowdhery et al., 2023; Touvron et al., 2023; Zeng et al., 2022), their ability to adapt to dynamic changes in data, tasks, and user preferences has become increasingly critical. Conventional training paradigms, which rely on static datasets for model development, are proving insufficient to address the rapidly evolving and dynamic nature of real-world information (Zhang et al., 2023b).

For instance, in 2020, the query "Who is the President of the United States?" would have yielded "Donald Trump" as the answer. However, the current answer is "Joe Biden." This exemplifies the ongoing challenge faced by LLMs: the need for continuous updating to ensure they reflect accurate and up-to-date knowledge.

Current approaches to model editing and knowledge updating typically involve augmenting the network architecture (Dong et al., 2022; Huang et al., 2022; Raunak and Menezes, 2022), introducing additional model parameters (Dai et al., 2023; Dong et al., 2022; Huang et al., 2022), or integrating external knowledge bases (Dai et al., 2023; Dong et al., 2022; Huang et al., 2022). These methods often necessitate more complex procedures than straightforward fine-tuning with new knowledge (Zhang et al., 2022; Li and Liang, 2021; Hu et al., 2021).

At present, direct fine-tuning of the model remains the predominant method for incorporating new knowledge.

During human cognitive development, individuals often encounter situations where new knowledge conflicts with their existing understanding. They usually remember both the new knowledge and the old knowledge simultaneously, and then often get confused, leading to contradictions that make it difficult to learn the new knowledge. If we directly modify the memory of old knowledge and original cognition, then the new knowledge to be learned will not conflict with the original cognition and knowledge, which makes it better to learn and absorb the new knowledge. For example, if people have been educated to believe that "the Earth is flat" since childhood, it would be challenging for them to accept the conflicting knowledge that "the Earth is round" when they become adults. Conversely, if they could directly modify their memory of the erroneous knowledge "the Earth is flat" to the correct knowledge "the Earth is round," it would be much simpler.

So how do we locate the position of old knowledge and then update it accurately? Our research has shown that when fine-tuning large language models, they tend to learn sentence structure, grammar, and style first, with knowledge being acquired last. Therefore, we control the variables to prevent the model from learning sentence structure and stylistic information.

Inspired by the above empirical observations and

(Ilharco et al., 2022)'s task arithmetic, we propose a novel paradigm of knowledge updating called **DFT** (Delicate Fine-Tuning). Specifically, DFT begins by using the large language model to predict and generate an answer, resulting in a data point. Next, DFT modifies only the key knowledge within the sentence, keeping the sentence structure and style intact, creating a new data point. We then fine-tune the model separately with both data points, recording the parameter changes. By comparing these parameter changes, we identify sections that exhibit similar changes in direction. These sections, representing aspects that are not relevant to the knowledge update, are discarded entirely.

We retain only the parameters exhibiting contrasting change directions, then compare their differences, rank those differences, and identify the top T % with the largest differences .Then update the top T % of parameters , where T is a predefined threshold ratio. The whole process is repeated iteratively until the model's knowledge update is complete.

This paper makes the following contributions:

- We propose a novel fine-tuning paradigm "**DFT** (Delicate Fine-Tuning)" for knowledge updating in large language models.
- Our experimental results show that **DFT** (Delicate Fine-Tuning) improves the knowledge updating performance across various fine-tuning methods and surpasses existing baselines in most cases.

2 Related Work

The knowledge update of large language models generally encompasses five approaches: model editing, meta-learning, fine-tuning, retrieval-augmented learning, and the addition of supplementary parameters(Yao et al., 2023; Zhang et al., 2024; Shi et al., 2024) .

Many necessitate the introduction of additional knowledge bases, neural network modules, or model parameters. This often leads to practical challenges, including increased model complexity and inference costs. Fine-tuning methods, even when updating a single piece of knowledge, can trigger a ripple effect, potentially affecting other knowledge and leading to catastrophic forgetting. A new model editing method is proposed in this paper, which eliminates a large number of irrelevant parameters by comparing old and new knowledge, and then identifies the most relevant parameters. By updating only the minimum number of parameters, this approach significantly reduces catastrophic forgetting.

Model editing

Model editing targets the internal mechanisms of LLMs, aiming to modify specific parameters and neurons to correct outputs based on knowledge-driven interventions (Meng et al., 2022a; Dai et al., 2022; Meng et al., 2022c; Santurkar et al., 2021; Geva et al., 2022). (Geva et al., 2021) discovered that the feed-forward network layers within transformers store key-value pairs associated with specific knowledge. (Meng et al., 2022a) employed a causal reasoning method to identify key neuron activations and update factual associations by modifying feed-forward weights. To facilitate large-scale knowledge editing, they introduced (Meng et al., 2022c), a method that directly updates thousands of memories within LLMs. (Gupta et al., 2023) enhanced knowledge updating by optimizing edit token selection and layer selection during the editing process. (Yu et al., 2023) utilized partitioned gradients to identify significant weights for unlearning biases in the model.

Hiyouga (hiyouga, 2023) developed the fastedit software framework, which enables convenient editing of models using causal reasoning. Zhang et al. (Zhang et al., 2024; Wang et al., 2023; Yao et al., 2023; Cheng et al., 2023; Mao et al., 2023; Zhang et al., 2023a) developed the EasyEdit software framework, which makes it easy to use a variety of methods for editing models.

Meta-learning

Meta-learning approaches aim to update knowledge within LLMs by adjusting their parameters based on predictions from a well-trained hypernetwork. This technique, investigated by (Sinitsin et al., 2019; Mitchell et al., 2021; De Cao et al., 2021), enables efficient knowledge updates without retraining the entire model. (Mitchell et al., 2021) introduced an auxiliary network with gradient decomposition, enabling efficient edits to LLMs based on a single input-output pair.(De Cao et al., 2021) proposed updating specific weights within a subset of modules using a hypernetwork with constrained optimization.

Fine-tuning

Fine-tuning has become a ubiquitous technique in NLP research, owing to the widespread adoption of pre-trained models for downstream tasks. Its intuitive nature and effectiveness in imparting new knowledge make it a valuable tool for model editing (Zhu et al., 2020; Zhang et al., 2022; Yao et al., 2023). Recent advancements in parameter-efficient fine-tuning methods, such as Prefix-Tuning (Li and Liang, 2021) and LoRA ((Hu et al., 2021)), have further enhanced its applicability to knowledge editing. (Zhang et al., 2022) proposed an adaptive fine-tuning strategy that dynamically adjusts the magnitude of parameter updates based on the importance of the weight matrix, thereby improving efficiency and adaptability. (Zhu et al., 2020) introduced a loss constraint that minimizes the impact on irrelevant knowledge during fine-tuning, preserving the integrity of the base model. Similarly, (Lee et al., 2022) explored large-scale continual learning for knowledge updating through regularized fine-tuning.

F-Learning (Ni et al., 2023) introduces a "forgetting before learning" paradigm to achieve forgetting of old knowledge and learning of new knowledge based on parametric arithmetic.

The addition of supplementary parameters

This approach involves injecting a small number of trainable parameters, representing new knowledge, into the LLM while keeping its original parameters frozen . This technique, explored by (Dong et al., 2022; Huang et al., 2022; Raunak and Menezes, 2022; Dai et al., 2023), allows for efficient knowledge injection without retraining the entire model. (Dong et al., 2022) proposed a lightweight feed-forward network that incorporates additional parameters specifically tailored to factual contexts, enabling knowledge generalization.(Huang et al., 2022) developed a model editor named Transformer-Patcher, which sequentially corrects errors in LLM outputs by adding and training a limited number of neurons within the transformer architecture.

Retrieve augmentation

These methods rely on an external knowledge base containing new or corrected information, aiming to amend the output of LLMs by incorporating retrieved knowledge relevant to the given prompt or question. This approach, explored by (Murty et al., 2022; Mitchell et al., 2022b; Li et al., 2022; Madaan et al., 2022), facilitates the integration of new knowledge into the model's responses.

(Mitchell et al., 2022b) propose a memory module that stores manual edits, enabling a classifier to retrieve and apply the relevant knowledge.(Madaan et al., 2022) leverage the memory of user feedback to generate prompts that guide LLMs toward more accurate responses. Alternatively, (Zheng et al., 2023) utilize in-context learning to revise LLM outputs by extracting demonstrations from a corpus based on similarity, eliminating the need for gradient calculations.

3 Task Definition

Here, we draw on certain concepts and formulas from (Ni et al., 2023).This paper addresses the task of knowledge updating in large language models (LLMs). Given a pre-trained model f_θ and a set of input-output knowledge pairs $K_{old} = (x_1, y_1), (x_2, y_2), ..., (x_i, y_i)$, the objective is to modify the model parameters θ to obtain a new model f_{θ^*} that generates a corresponding set of updated input-output pairs $K_{new} = (x_1, y_1^{new}), (x_2, y_2^{new}), ..., (x_i, y_i^{new})$. Here, i represents the number of knowledge pairs to be updated.

Following the definition in (Yao et al., 2023), we can formally express this process and its objective as:

$$f_{\theta^*}(x_i) = \begin{cases} y_i^{new} & \text{if } x_i \in N(x_i) \\ f_\theta(x_i) & \text{if } x_i \in other \end{cases} \quad (1)$$

where $N(x_i)$ represents x_i itself and its equivalent neighbourhood.

The knowledge update task aims to modify the model's responses only for x_i and its equivalent domain $N(x_i)$, where $N(x_i)$ represents the neighborhood of x_i encompassing semantically equivalent instances. The goal is to update the answers associated with x_i and its equivalent domain without affecting the responses to other out-of-scope knowledge.

The effectiveness of knowledge updating is evaluated based on the following three metrics:

a. Reliability

Measured as the average accuracy of the updated model f_{θ^*} on the new knowledge. This metric assesses the effectiveness of the update process itself.

For example, the answer to the question "Who is the President of the US?" should be updated from "Donald Trump" to "Joe Biden" after knowledge updating.

b. Generalization

Evaluated by the average accuracy of f_{θ^*} on examples drawn uniformly from the equivalence neighborhood $N(x_i)$. This metric assesses the ability of the model to generalize the update to semantically equivalent inputs. For example, the answer to the question "Who holds the position of the President of the US?" should also be updated from "Donald Trump" to "Joe Biden".

c. Locality

Assessed by the proportion of predictions from the updated model f_{θ^*} that remain unchanged compared to the pre-update model f_θ on irrelevant examples. This metric evaluates the ability of the model to preserve the original knowledge base while updating specific knowledge. For example, the answer to the question "'You're fired!' is the catchphrase of which celebrity?" should remain unchanged as "Donald Trump" after the update.

4 Proposed method: DFT

This section details our proposed approach for knowledge updating in LLMs. Departing from methods that rely on external knowledge bases or additional parameters, our method leverages a full fine-tuning strategy. The process is structured in two distinct stages:

4.1 Locate the parameters associated with the old knowledge

Supervised fine-tuning (SFT) on a designated dataset enables us to identify the direction of parameter alignment with the desired knowledge. This alignment is reflected in the variations observed in the model's parameters during the training process. Within this framework, we define incremental parameters, denoted as θ_Δ, as knowledge parameters for a given large language model f_θ and its parameters θ. These knowledge parameters are computed as follows:

$$\theta_\Delta = \mathrm{FT}\{\theta, \mathrm{K}\} - \theta \tag{2}$$

where FT is the operation of supervised fine-tuning, while K, θ refer to the dataset of knowledge and the parameters of the original model f_θ, respectively.

Analogously, we initially fine-tune the model f_θ on a dataset comprising the model's original knowledge. Subsequently, we subtract the original model parameters θ from the parameters obtained after fine-tuning to derive the knowledge parameters θ_Δ^{old}, representing the learned original knowledge. This calculation is expressed as:

$$\theta_\Delta^{old} = \mathrm{FT}\{\theta, \mathrm{K}_{\mathrm{old}}\} - \theta \tag{3}$$

where K_{old} refers to a dataset composed of the model's original knowledge. The related work in (Ilharco et al., 2022) considers that subtracting the parameters θ_Δ^{old} from θ can assist the model f_θ to forget this part of old knowledge:

$$\theta' = \theta - \lambda\theta_\Delta^{old} \tag{4}$$

where λ is a hyper-parameter to control the rate of forgetting. This process yields a new model, $f_{\theta'}$, with parameters θ', which exhibits reduced retention of the original knowledge compared to the initial model f_θ. The forgetting operation may have a destructive effect on the normal knowledge of the model.

However, we believe that θ_Δ^{old} also contains other information such as sentence structure, grammar, and style, which requires further processing to accurately pinpoint the old knowledge.

Then, we re-fine-tune the model f_θ on a dataset containing new knowledge, and then subtract the parameters θ of the original model f_θ from model's parameters after fine-tuning to obtain the knowledge parameters θ_Δ^{new} indicating the new knowledge, as follows:

$$\theta_\Delta^{new} = \mathrm{FT}\{\theta, \mathrm{K}_{\mathrm{new}}\} - \theta \tag{5}$$

where K_{new} refers to a dataset composed of new knowledge .

Then, we compare θ_Δ^{old} and θ_Δ^{new}, discarding all elements with the same sign. Then, we select the top T % of elements in θ_Δ^{new} with the largest difference from θ_Δ^{old}. T is a predefined threshold ratio.The discarded parts are not subject to parameter updates,and only the final retained parts will be used for parameter updates,as follows:

$$\theta_\Delta^{core} = \mathrm{f}\{\theta_\Delta^{\mathrm{new}}, \theta_\Delta^{\mathrm{old}}, \mathrm{T}\} \tag{6}$$

θ_Δ^{core} represents the retained key parameters, which are used for parameter updates.All other parameters remain unchanged.

4.2 Learning new knowledge by updating only the most relevant parameters

We define the process of learning new knowledge as follows:

$$\theta^* = \theta + \lambda\theta_{\Delta}^{core} \tag{7}$$

where λ is a hyper-parameter to control the rate of learning. We repeat the processes outlined in equations (3), (5), (6), and (7) until the model's output reflects the new knowledge. Now we gain a new model f_{θ^*} with its parameters θ^*, which has forgotten the old knowledge compared to f_θ. It learns only the new knowledge, avoiding any other information, preventing catastrophic forgetting caused by style changes and the like.

5 Experiments

5.1 Datasets

Our experiments employ one widely used datasets: (Levy et al., 2017). ZsRE is a Question Answering (QA) dataset that leverages question rephrasings generated via back-translation to represent the equivalence neighborhood. Following the experimental setup outlined in (Yao et al., 2023), we utilize the evaluation (eval) and edit sets of these datasets, comprising 19,085 data points. To facilitate knowledge update, we partition both datasets into sets of old knowledge and new knowledge. For instance, in ZsRE, a typical knowledge update scenario involves modifying the answer from "Los Angeles" to "New Orleans", as illustrated in the following example:

The old knowledge:

{**"instruction"**: "What city did Marl Young live when he died?", **"input"**: "", **"output"**: "Los Angeles" }

The new knowledge:

{**"instruction"**: "What city did Marl Young live when he died?", **"input"**: "", **"output"**: "New Orleans" }

5.2 Baselines

We compare DFT against direct fine-tuning **FT-L** with an additional KL divergence loss (Meng et al., 2022a). We also compare DFT to other model editors, including GPT-style editors based on causal tracing: **ROME** (Meng et al., 2022a), **MEMIT** (Meng et al., 2022c).ROME ((Meng et al., 2022a)) is a method that updates specific factual associations through causal intervention. MEMIT ((Meng et al., 2022c)) is a method known for its effectiveness in directly updating large-scale memories within LLMs.

5.3 Completion details

For our experiments, we employ QWEN1.5-7B and Mistral-7B as the base models. The primary focus of our evaluation is the ability to update old knowledge with new knowledge. To maintain output consistency, we utilize the greedy decoding strategy during testing. Our experiments were conducted on a hardware platform comprising 8 x A800-80G GPUs.

5.4 Experimental results

Table 1 presents the experimental results, our DFT method consistently outperforms other baselines in most cases. FT-L performs the worst, possibly due to the impact of the KL divergence loss on the model's update. As our original model has already acquired a substantial amount of old knowledge, learning new knowledge poses greater challenges. ROME employs causal analysis to identify knowledge embedded within specific MLP layers and modifies the entire matrix using least squares approximation. It operates under the assumption that the MLP serves as the primary module for knowledge storage , and at each iteration, it injects a single piece of knowledge into the MLP through a Lagrangian remainder.In the experiments, ROME achieved relatively good results. Similarly, MEMIT based on the assumption that the FFN functions as a key-value store for knowledge, directly manipulates the parameters of specific layers using least squares approximation. In contrast to ROME, which updates a single layer, MEMIT is a multi-layer update algorithm that supports the simultaneous update of hundreds or thousands of facts. In the experiments, the difference between MEMIT and ROME was minimal. F-Learning outperforms MEMIT and ROME to some extent, while the DFT method demonstrates superior performance compared to F-Learning.

5.5 Ablation study

Table 1 shows that DFT method outperforms F-Learning (Ni et al., 2023). We analyze that this may be because F-Learning first uses the old data for fine-tuning in an attempt to forget old knowledge. However, the data contains a large amount of information, and what is forgotten may not necessarily

Dataset	Editor	Mistral-7B			QWEN1.5-7B		
		Reliability	Generality	Locality	Reliability	Generality	Locality
ZsRE	FT-L	58.19	51.48	80.43	56.45	50.72	81.19
	ROME	84.21	79.98	78.12	84.37	79.31	78.37
	MEMIT	79.36	78.82	86.78	83.25	80.39	87.24
	F-Learning$_{\text{FT}}$	84.85	80.15	85.07	85.64	81.85	88.93
	DFT$_{\text{FT}}$	85.41	83.87	91.27	89.78	86.26	92.19

Table 1: Results on three metrics of the one datasets based on QWEN1.5-7B and Mistral-7B.

be the old knowledge. It could be sentence structure, grammar, style, or other information, meaning it might not truly forget the old knowledge. Our method has an advantage in this regard, as it compares the new and old knowledge and highlights the key differences.

5.6 Updating with LoRA

Within this experimental framework, our approach involves simultaneous knowledge updating via full fine-tuning (or LoRA) in a single training process. We formally define this LoRA integrated approach as follows:

$$\theta_{\Delta}^{old} = \text{LoRA}\{\theta, \text{K}_{\text{old}}\} - \theta, \tag{8}$$

$$\theta_{\Delta}^{new} = \text{LoRA}\{\theta, \text{K}_{\text{new}}\} - \theta \tag{9}$$

$$\theta_{\Delta}^{core} = \text{f}\{\theta_{\Delta}^{\text{new}}, \theta_{\Delta}^{\text{old}}\} \tag{10}$$

$$\theta^{*} = \theta + \lambda\theta_{\Delta}^{core} \tag{11}$$

where LoRA represents the operation of supervised fine-tuning utilizing the LoRA technique . θ^* is noted as the parameters of the edited model f_{θ^*} which has completed the knowledge updating.

As presented in Table 1, the experimental results indicate that knowledge updating using LoRA outperforms full fine-tuning in certain instances. This improvement can be attributed to the parameter-efficient nature of LoRA-based knowledge forgetting, enabling more efficient learning and adaptation.

Empirical evidence from our experiments suggests that updating the model parameters through LoRA adaptation effectively approximates the performance achieved by full fine-tuning.

We hypothesize that this observation stems from the distributed nature of knowledge encoding across multiple model parameters. LoRA modifies the patterns and relationships associated with the old knowledge embedded within the attention structure, which represents an implicit knowledge representation.

Table 2: Results on three metrics of the zsRE dataset based on BLOOM-7B.

Editor	Metric		
	Reliability	Generality	Locality
Original model	28.02	27.95	/
LoRA	29.32	29.31	77.32
F-Learning$_{\text{LoRA}}$	29.28	29.07	77.44
DFT$_{\text{LoRA}}$	**30.38**	**31.02**	**79.64**
Full-FT	44.32	43.72	63.94
F-Learning$_{\text{FT}}$	44.87	43.95	69.53
DFT$_{\text{FT}}$	**45.83**	**44.64**	**72.15**

5.7 Adaptability testing

To further assess the adaptability of our proposed method, we conducted experiments on the zsRE dataset using BLOOM-7B as the base model. We maintained the same experimental settings as previously described. The results, presented in Table 2, demonstrate the continued effectiveness of DFT.

5.8 Time testing

To evaluate the efficiency of our proposed DFT method, we compared the editing time of various knowledge updating and model editing methods for different edit sizes. Employing LLAMA2-7B as our base model, we present the results in Table 3.

Analysis of the results in Table 3 reveals that fine-tuning-based methods consistently exhibit significantly lower editing times compared to locate-based methods. This disparity can be attributed to the increased complexity and time requirements associated with locating specific neurons and parameters in locate-based methods. Furthermore, ROME's limitation to single-datapoint edits, in contrast to the batch editing capabilities of other methods, further diminishes its efficiency. Among fine-tuning-based methods, FT-c demonstrates faster optimization due to its norm constraint.

Editor	**1 edit**		**10 edits**		**100 edits**	
	zsRE	**COUNTERFACT**	**zsRE**	**COUNTERFACT**	**zsRE**	**COUNTERFACT**
FT-L	0.61(s)	0.57(s)	5.98(s)	5.73(s)	58.28(s)	57.08(s)
ROME	2.76(s)	2.46(s)	27.9(s)	24.32(s)	285.23(s)	242.21(s)
MEMIT	612(s)	606(s)	6231(s)	6193(s)	61831(s)	61631(s)
Full-FT	0.78(s)	0.74(s)	7.92(s)	7.43(s)	76.72(s)	75.11(s)
DFT$_{\mathrm{FT}}$	**1.49(s)**	**1.42(s)**	**11.98(s)**	**11.21(s)**	**120.92(s)**	**118.87(s)**

Table 3: Editing time for 1 edit, 10 edits, 100 edits of the two dataset based on LLAMA2-7B.Run ROME with FastEdit.Run MEMIT with EasyEdit

DFT method, while requiring multiple backward passes and comparisons as a multi-stage knowledge updating approach, necessitates updating only a limited set of parameters. Consequently, DFT exhibits an editing time approximately twice that of Full-FT, yet remains notably fast and convenient.

Further acceleration of supervised fine-tuning can be achieved through the utilization of deepspeed or other analogous optimization techniques.

5.9 Parametric analysis of updating knowledge

DFT knowledge updating method hinges on the precise identification of knowledge-related parameters within the model. From an interpretability perspective, this approach allows us to pinpoint specific parameters containing the desired knowledge, enabling targeted updates. Furthermore, we conducted an in-depth analysis of the parameter distribution and its modifications within the LLMs.

Analysis reveals that parameter modifications in the MLP layers are more pronounced than those observed in the attention layers. This observation suggests that knowledge is primarily encoded within the MLP layers of the model.

6 Conclusion

This paper introduces a novel paradigm for knowledge updating during supervised fine-tuning, termed DFT (Differential Fine-Tuning). DFT leverages parametric arithmetic to pinpoint the location of existing knowledge and facilitates the acquisition of new knowledge, effectively resolving potential contradictions between old and new information.

Experimental evaluations conducted on the zsRE dataset demonstrate the superior performance of our proposed method compared to other baselines in most scenarios.

7 Limitations

While the proposed DFT paradigm enhances the efficacy of fine-tuning methods for updating knowledge in large language models, it incurs an increase in computational overhead due to the incorporation of multiple backward passes.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Siyuan Cheng, Bozhong Tian, Qingbin Liu, Xi Chen, Yongheng Wang, Huajun Chen, and Ningyu Zhang. 2023. Can we edit multimodal large language models? *arXiv preprint arXiv:2310.08475*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

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 (Volume 1: Long Papers)*, pages 8493–8502.

Damai Dai, Wenbin Jiang, Qingxiu Dong, Yajuan Lyu, and Zhifang Sui. 2023. Neural knowledge bank for pretrained transformers. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 772–783. Springer.

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.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5937–5947.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45.

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.

Anshita Gupta, Debanjan Mondal, Akshay Krishna Sheshadri, Wenlong Zhao, Xiang Lorraine Li, Sarah Wiegreffe, and Niket Tandon. 2023. Editing commonsense knowledge in gpt. *arXiv preprint arXiv:2305.14956*.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2022. Aging with grace: Lifelong model editing with key-value adaptors.

hiyouga. 2023. Fastedit: Editing llms within 10 seconds. https://github.com/hiyouga/FastEdit.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2022. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Kyungjae Lee, Wookje Han, Seung-won Hwang, Hwaran Lee, Joonsuk Park, and Sang-Woo Lee. 2022. Plug-and-play adaptation for continuously-updated qa. *arXiv preprint arXiv:2204.12785*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2022. Large language models with controllable working memory. *arXiv preprint arXiv:2211.05110*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve gpt-3 after deployment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2833–2861.

Shengyu Mao, Ningyu Zhang, Xiaohan Wang, Mengru Wang, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2023. Editing personality for llms. *arXiv preprint arXiv:2310.02168*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022b. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2022c. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. In *International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022a. Fast model editing at scale. In *International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022c. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR.

Shikhar Murty, Christopher D Manning, Scott Lundberg, and Marco Tulio Ribeiro. 2022. Fixing model bugs with natural language patches. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11600–11613.

Shiwen Ni, Dingwei Chen, Chengming Li, Xiping Hu, Ruifeng Xu, and Min Yang. 2023. Forgetting before learning: Utilizing parametric arithmetic for knowledge updating in large language models. *arXiv preprint arXiv:2311.08011*.

Vikas Raunak and Arul Menezes. 2022. Rank-one editing of encoder-decoder models. *arXiv preprint arXiv:2211.13317*.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. 2021. Editing a classifier by rewriting its prediction rules. *Advances in Neural Information Processing Systems*, 34:23359–23373.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. 2024. Continual learning of large language models: A comprehensive survey. *arXiv preprint arXiv:2404.16789*.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitry Pyrkin, Sergei Popov, and Artem Babenko. 2019. Editable neural networks. In *International Conference on Learning Representations*.

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

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, et al. 2023. Easyedit: An easy-to-use knowledge editing framework for large language models. *arXiv preprint arXiv:2308.07269*.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*.

Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, and Heng Ji. 2023. Unlearning bias in language models by partitioning gradients. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6032–6048.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. In *International Conference on Learning Representations*.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. 2024. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*.

Ningyu Zhang, Jintian Zhang, Xiaohan Wang, Honghao Gui, Kangwei Liu, Yinuo Jiang, Xiang Chen, Shengyu Mao, Shuofei Qiao, Yuqi Zhu, Zhen Bi, Jing Chen, Xiaozhuan Liang, Yixin Ou, Runnan Fang, Zekun Xi, Xin Xu, Lei Li, Peng Wang, Mengru Wang, Yunzhi Yao, Bozhong Tian, Yin Fang, Guozhou Zheng, and Huajun Chen. 2023a. Knowlm technical report.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2022. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*.

Zihan Zhang, Meng Fang, Ling Chen, Mohammad Reza Namazi Rad, and Jun Wang. 2023b. How do large language models capture the ever-changing world knowledge? a review of recent advances. In *Empirical Methods in Natural Language Processing*.

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? *arXiv preprint arXiv:2305.12740*.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.