How Well Can Knowledge Edit Methods Edit Perplexing Knowledge?

Anonymous ACL submission

Abstract

As large language models (LLMs) are widely deployed, targeted editing of their knowledge has become a critical challenge. Recently, advancements in model editing techniques, such as Rank-One Model Editing (ROME) (Meng et al., 2022a), have paved the way for updating LLMs with new knowledge. However, the efficacy of these methods varies across different types of knowledge. This study investigates the capability of knowledge editing methods to incorporate new knowledge with varying degrees of "perplexingness", a term we use to describe the initial difficulty LLMs have in understanding new concepts. We begin by quantifying the "perplexingness" of target knowledge using pre-edit conditional probabilities, and assess the efficacy of edits through post-edit conditional probabilities. Utilizing the widely-used COUNTERFACT dataset (Meng et al., 2022a), we find significant negative correlations between the "perplexingness" of the new knowledge and the edit efficacy across all 12 scenarios. To dive deeper into this phenomenon, we introduce a novel dataset, HIERARCHYDATA, consisting of 99 hyponym-hypernym pairs across diverse categories. Our analysis reveal that more abstract concepts (hypernyms) tend to be more perplexing than their specific counterparts (hyponyms). Further exploration into the influence of knowledge hierarchy on editing outcomes indicates that knowledge positioned at higher hierarchical levels is more challenging to modify in some scenarios. Our research highlights a previously overlooked aspect of LLM editing: the variable efficacy of editing methods in handling perplexing knowledge. By revealing how hierarchical relationships can influence editing outcomes, our findings offer new insights into the challenges of updating LLMs and pave the way for more nuanced approaches to model editing in the future.

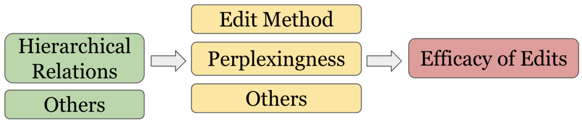


Figure 1: The whole structure: what influence the edit efficacy.

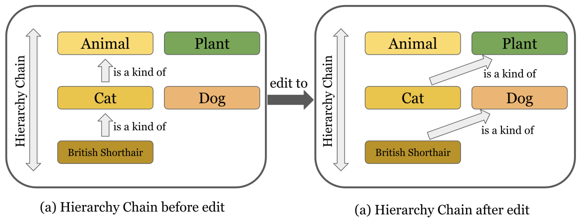


Figure 2: Two examples in our HIERARCHYDATA dataset, along a hierachy chain.

1 Introduction

Large language models (LLMs) possess the capability to predict factual statements about the world, and recent advancements have enabled the editing of the factual knowledge embedded within these models. Such editing not only aids in rectifying inaccuracies within the large language models but also serves as a valuable approach for comprehending the complex mechanisms of these extensive, often opaque, neural networks. Among the various methodologies for knowledge editing, Rank-One Model Editing (ROME) (Meng et al., 2022a) and Mass-Editing Memory In a Transformer (MEMIT) (Meng et al., 2022b) stand out as notable ones.

Knowledge editing methods show great potential for controlling LLMs. As researchers deploy the knowledge editing methods, it is increasingly important to know the boundary of the capacities of the knowledge editing methods.

In this paper, we try to answer this question: *Is the perplexing knowledge more difficult to edit?*

We explore knowledge editing methods within the context of their ability to update new information, with a specific emphasis on the notion of "perplexingness". This concept serves to gauge the initial challenge faced by LLMs when encountering new or unfamiliar information. By assessing the "perplexingness" of the target knowledge through pre-edit conditional probabilities and evaluating the efficacy of these edits through post-edit conditional probabilities, our research endeavors to illuminate the intricate dynamics involved in the process of updating knowledge within LLMs.

Leveraging the COUNTERFACT dataset, we investigate various knowledge editing approaches—including Fine-Tuning (FT), Low-Rank Adaptation (LoRA), ROME, and MEMIT—applied to models such as GPT2-Large, GPT2-XL, and GPT-J (6B). Our research identifies significant negative correlations between the "perplexingness" of new knowledge and the edit efficacy across a spectrum of scenarios.

To deepen our comprehension of the elements that influence "perplexingness", we introduce the HIERARCHYDATA dataset, comprised of 99 hyponym-hypernym pairs spanning a variety of categories. Our analysis demonstrates that hierarchical relations significantly affect the knowledge "perplexingness" in the models. Abstract concepts (hypernyms) tend to present a higher level of "perplexingness" compared to their more specific counterparts (hyponyms). Additionally, we analyze the relationships between "perplexingness" and edit efficacy, as well as the relationships between hierarchical relations and edit efficacy.

Figure 1 illustrates the overall structure of factors that may influence the efficacy of edits. In this paper, we contribute a novel perspective on LLM editing by highlighting how the "perplexingness" of knowledge affects the efficacy of edits. Additionally, we develop the HIERARCHYDATA dataset, which is the first to consider hierarchical relations when editing knowledge in models. Our findings indicate that hierarchical relations influence "perplexingness." As we continue to unravel the complexities of editing LLMs, this research serves as a foundation for future endeavors aimed at refining and enhancing the adaptability of these knowledge edit methods.

2 Related Work

2.1 Knowledge Edit Methods

Various approaches have been developed to modify the knowledge embedded in large language models. Rank-One Model Editing (ROME) (Meng et al., 2022a) updated feed-forward weights to alter specific factual associations. MEMIT (Meng et al., 2022b) allowed for the incorporation of numerous memories into a language model. Low-Rank Adaptation (LoRA) (Hu et al., 2021) maintains pre-trained weights while using trainable decomposition matrices for efficient, targeted updates without altering the original weights. Model Editor Networks with Gradient Decomposition (MEND) (Mitchell et al., 2021) utilized a single targeted input-output pair for quick, localized adjustments in a pre-trained model's behavior. Other notable methods include editing specific knowledge neurons (Dai et al., 2021), employing hyper-networks (De Cao et al., 2021), and applying linear transformations (**?**). These techniques have demonstrated impressive efficacy in modifying knowledge in large language models. There are also works that apply model editing to gain novel insights about the model interpretability (Niu et al., 2024; Hase et al., 2024). However, the performance of the model editing techniques is typically assessed in a broad context. We delve into whether model editing methods are applicable to knowledge with different "perplexingness". We specifically examine the impact of the conditional probability of the target words for editing and the hierarchical relationships among words on the overall performance of these editing techniques.

2.2 Limitation of Knowledge Edit Methods

Recent research has identified certain limitations in the methods used for editing large language models. Firstly, some studies have concentrated on the specificity of edits, developing new metrics and benchmarks for evaluation. Hoelscher-Obermaier et al. (2023) enhanced existing benchmarks by introducing a dynamic component and proposed a KL divergence-based metric for measuring specificity. Li et al. (2023) introduced an evaluation protocol and a question-answer dataset designed to assess edit specificity.

Secondly, the consistency of edits has been another focal point. Zhong et al. (2023) devised a multi-hop question benchmark to test whether models can correctly respond to questions affected by edited facts. Wu et al. (2023) examined knowledge editing through reasoning and cross-lingual knowledge transfer. Ma et al. (2024) looked into if edited LLMs can behave consistently resembling communicative AI in realistic situations. Li et al. (2023) also offered a protocol to evaluate edit consistency, while Onoe et al. (2023) investigated the ability of Large Language Models to infer and propagate injected facts. Rosati et al. (2024) introduced a long-form evaluation protocol, assessing the effects of model editing beyond the immediate "next token"; we consider the effects of the model editing methods that can be assessed at the next token.

Thirdly, the nature of the edited knowledge has been scrutinized. Gupta et al. (2023) specifically evaluated editing methods on commonsense knowledge statements, as opposed to encyclopedic knowledge. Ma et al. (2024) examined which knowledge features are correlated with the performance and robustness of editing.

While these studies cover various aspects, only a few delve into the impact of the type of knowledge being edited. In this paper, we explore how the "perplexingness" of the knowledge and the hierarchical relations among words influence the efficacy of editing methods in large language models.

3 Model Edit Methods

For a knowledge edit task, we represent each fact as a knowledge tuple $t = (s, r, o)$. For each fact, we want to insert a new knowledge tuple $t = (s, r, o^*)$. Recent studies explored different ways to edit knowledge, including Fine-Tuning (FT), Low-Rank Adaptation (LoRA), Rank-One Model Editing (ROME) and Mass-Editing Memory In a Transformer (MEMIT).

Fine-Tuning (FT) This traditional method involves applying Adam optimization (Kingma and Ba, 2014) with early stopping at one layer to edit knowledge. It directly adjusts the weights of the model through backpropagation, affecting the entire layer where the edit is applied.

Low-Rank Adaptation (LoRA) (Hu et al., 2021) Unlike FT, LoRA freezes the pre-trained model weights and introduces trainable rank decomposition matrices at each layer of the Transformer. This method significantly reduces the number of trainable parameters needed for editing, focusing on a more efficient and targeted update mechanism without altering the original model weights directly.

Rank-One Model Editing (ROME) (Meng et al., 2022a) ROME specifically targets the feed-forward weights within the Transformer's MLP layers, viewing them as associative memory. By computing and inserting a key-value pair (k, v) into this memory through a constrained least-squares problem, ROME offers a precise and efficient way to update factual knowledge. This method focuses on modifying specific factual associations with minimal impact on the overall model.

Mass-Editing Memory In a Transformer (MEMIT) (Meng et al., 2022b) Building on the direct editing approach of ROME, MEMIT is designed for large-scale updates, capable of handling thousands of associations. It directly targets transformer module weights identified as causal mediators of factual knowledge recall, aiming for a broad and scalable editing solution.

In summary, while FT and LoRA focus on general model adjustments with varying degrees of parameter freedom, ROME and MEMIT offer more targeted and efficient approaches to knowledge editing, with MEMIT specifically designed for mass-editing scenarios.

4 Data and tool

4.1 Data

CounterFact (Meng et al., 2022a) is a dataset designed to assess counterfactual edits in language models. It includes a collection of challenging incorrect facts (s, r, o^*) and the accurate facts (s, r, o). In this context, s represents the subject, r delineates the relation, and o corresponds to the object. The prompt consists of predetermined templates based on r, which are then completed with s. For instance, in the statement "A British Shorthair is a kind of cat", "A British Shorthair" represents s, "is a kind of" signifies r, and "cat" is denoted by o.

HierarchyData encompasses a series of both challenging incorrect facts, represented as (s, r, o^*), and their corresponding accurate facts, denoted as (s, r, o). It also draws upon a curated collection of hierarchy chains, as illustrated in Figure 2. Here, s signifies the subject and o the object, both selected from the hierarchy chains. The relation r consistently adopts the "is a kind of" schema, emphasizing hierarchical connections. This dataset is organized into two hierarchical levels: specific level (hyponyms), and abstract level (hypernyms). An example of such a hierarchy chain is "British Shorthair → Cat → Animal" from which we can infer the specific relationship "A British Shorthair is a kind of cat" and the more abstract relationship "A cat is a kind of animal." The focal point of our investigation is to assess the performance of editing methodologies on these two distinct types of facts within the hierarchical framework, exploring whether the level of abstraction within the hierarchy affects editing efficacy. To this end, we modify the objects of these facts individually, generating altered facts such as "A British Shorthair is a kind of dog" and "A cat is a kind of plant" to test the efficacy of edit methods against the backdrop of hierarchical data complexity. The HIERARCHY DATA dataset includes approximately 99 such chains, culminating in a corpus of 198 facts targeted for editing analysis. This structured approach facilitates explorations into the role of hierarchical relations in the adaptability and accuracy of language model editing processes.

4.2 Tool

We employ four knowledge editing method: FT, LoRA, ROME and MEMIT, sourced from the EasyEdit repository (Wang et al., 2023) to conduct our experiments.

5 Experiment

The experiments conducted in this study are designed to evaluate the efficacy of several knowledge editing methods, including FT, LoRA, ROME, and MEMIT. Our approach involves the substitution of a knowledge tuple, denoted as (s, r, o^*), for the existing tuple (s, r, o). In this context, s represents the subject, r delineates the relation, and o corresponds to the object. This analysis is carried out using three distinguished large language models: GPT2-Large, GPT2-XL, and GPT-J (6B).

5.1 "Perplexingness" of Knowledge

First, we want to define perplexing knowledge. People find knowledge perplexing when they cannot understand it. So we define the perplexing knowledge as the knowledge that the model cannot easily understand. Therefore, we define perplexing knowledge as knowledge that the model cannot easily understand. We quantify the "perplexingness" of knowledge as the conditional probabilities of new targets prior to editing. For easier comparison, we use the negative log form of the probability: the higher the value, the lower the probability, and the more perplexing the model finds the new knowledge.

It is important to note that we define 'perplexingness' based on the model's poor understanding of the knowledge, not its complexity. Even if a piece of knowledge is complex, if it is well known to the model due to effective pre-training, we do not consider it perplexing to the model.

Second, we evaluate the edit performance by its efficacy. Here, the efficacy of edits is defined as the conditional probabilities of new targets after the edit. We also express these conditional probabilities in the form of negative logarithms for more intuitive data interpretation. A lower "Efficacy" value indicates greater edit efficacy. The formulas are presented as follows:

$$\text{Perplexingness} = -\log P_{\text{pre-edit}}[o^*|s, r], \quad (1)$$

$$\text{Efficacy} = -\log P_{\text{post-edit}}[o^*|s, r]. \quad (2)$$

The investigation into the perplexing knowledge and the efficacy of edits employs the COUNTERFACT dataset. For each large language model, a total of 2,000 data groupings were analyzed.

	FT	LoRA	ROME	MEMIT
GPT2-large	0.482*	0.236*	0.288*	0.640*
GPT2-XL	0.158*	0.324*	0.259*	0.486*
GPT-J	0.204*	0.203*	0.062*	0.076*

Table 1: COUNTERFACT data Pearson correlation betweeen "perplexingness" and edit efficacy (∗ indicates corresponding entry has p-value below 0.05).

Correlations between "perplexingness" and edit efficacy We chart the "perplexingness"(pre-edit probabilities of the new target) against the efficacy (post-edit probabilities of new target). The scatter plots (see Appendix A) generated from this analysis provide a visual representation of the relationship between pre-edit and post-edit probabilities for the new target outcomes. The left panel of Figure 3 provides an example of these scatter plots, showcasing the application of MEMIT on GPT2-XL. This visulaization clearly illustrates a positive correlation between "perplexingness" of knowledge and efficacy of edits.

Correlations are significant To quantify this relationship, Pearson correlation coefficients are computed and are presented in Table 1. Additionally, to assess the statistical significance of these correlations, p-values are calculated. Entries corresponding to p-values falling below the significance threshold of 0.05 are marked with $*$ within the table.

It is observed that all the coefficients' p-values are beneath the 0.05 threshold, thereby indicating a statistically significant correlation between "perplexingness" and edit efficacy. **This means that when a model finds new knowledge very perplexing, it is difficult to incorporate this knowledge into the model.** Similarly, a person might be resistant to learning something they find hard to understand.

Furthermore, the analysis reveals that certain scenarios exhibit high Pearson coefficients, such as the application of MEMIT to the GPT-2 large model. This variance could stem from the possibility that different models encode "perplexingness" in distinct manners, and that editing methods may interact with this "perplexingness" uniquely.

Correlation is in the new knowledge but not the original knowledge Our analysis specifically focuses on the conditional probabilities of newly introduced knowledge (s, r, o^*), as opposed to the original knowledge (s, r, o) that stored in the language models. Early efforts to evaluate the conditional probabilities of the original knowledge did not show any significant correlation with the editing process, suggesting a mostly arbitrary relationship.

5.2 Hierarchical relations

To enhance our understanding of the factors contributing to "perplexingness", we introduce a dataset named HIERARCHYDATA. This dataset is aimed at investigating whether hierarchical relations between words can affect "perplexingness", subsequently influencing the edit efficacy.

Significantly higher "perplexingness" of higher hierarchy level knowledge Do hierarchical relations affect "perplexingness"? We divide the HIERARCHYDATA into two groups: hypernyms (abstract concepts) and hyponyms (specific concepts). For example, a statement like "A British Shorthair is a kind of cat" represents a specific level, while "A cat is a kind of animal" exemplifies an abstract level. To investigate the effect of hierarchical relations on "perplexingness," we analyze these two groups. The box plots are included in Appendix D. We conduct t-tests for two independent samples to determine if the mean "perplexingness" of the specific level is statistically lower than that of the abstract level. The results of the t-tests are detailed in Table 2, with all values demonstrating statistical significance. Our findings indicate that **knowledge on a higher hierarchical level (more abstract) is associated with greater "perplexingness" for the models**. This suggests that hierarchical relations are a factor affecting knowledge "perplexingness" for language models.

GPT2-Large	GPT2-XL	GPT-J
0.00728^*	0.00605^*	$1.330e-06^*$

Table 2: Comparative analysis of "perplexingness" in HIERARCHYDATA: t-test results for specific vs. abstract level distributions ($*$ indicates corresponding entry has p-value below 0.05).

	FT	LoRA	ROME	MEMIT
GPT2-large	0.893^*	0.886^*	0.167^*	0.575^*
GPT2-XL	0.860^*	0.856^*	0.148^*	0.381^*
GPT-J	0.454^*	0.755^*	0.078	-0.019

Table 3: HIERARCHYDATA Pearson correlation betweeen "perplexingness" and edit efficacy ($*$ indicates corresponding entry has p-value below 0.05)

Correlations between "perplexingness" and edit efficacy Next, we aim to determine if the correlation between "perplexingness" and edit efficacy also holds for the HIERARCHYDATA dataset. We employ the same method to analyze HIERARCHYDATA as analyzing COUNTERFACT, focusing on the Pearson correlation coefficient between "perplexingness" and edit efficacy. The right panel of Figure 3 provides one of the scatter plots (see Appendix B for other plots), showcasing the application of MEMIT on GPT2-XL. We also calculate the Pearson coefficients, with the results presented in Table 3. In this table, p-values below 0.05 are marked with $*$, indicating statistical significance. Our analysis reveals a consistent trend: an increase in "perplexingness" correlates with poorer efficacy of edits (higher negative log conditional probability). This pattern holds true across all scenarios, except when applying the ROME and MEMIT techniques to the GPT-J model.

Relationships between hierarchical relations and edit efficacy Additionally, we want to determine if hierarchical relations within the knowledge ultimately affect the edit efficacy. Box plots (see Appendix C) are constructed to visually compare

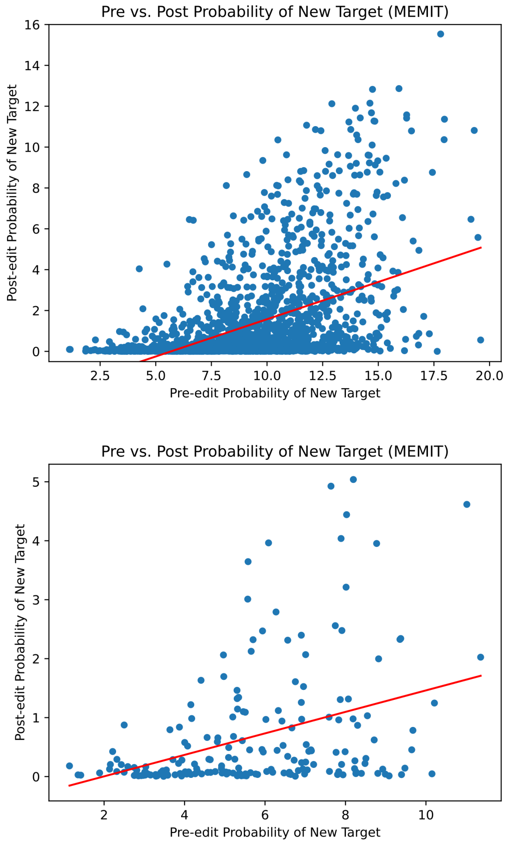


Figure 3: Pre vs. post probability of new knowledge (MEMIT on GPT2-XL). a.COUNTERFACT(left). b.HIERARCHYDATA(right).

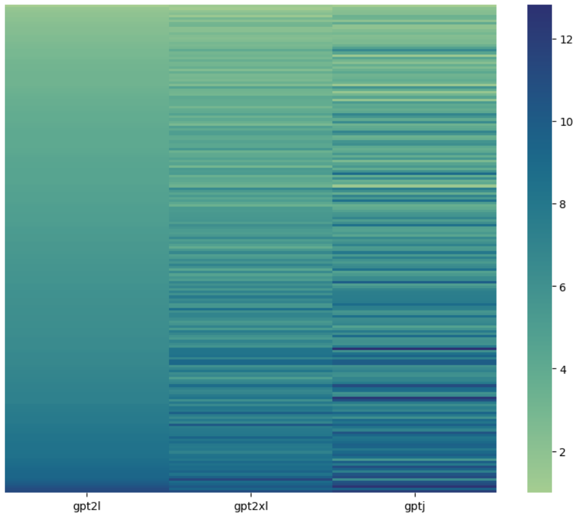


Figure 4: Same knowledge "perplexingness" in different models (HIERARCHYDATA).

	FT	LoRA	ROME	MEMIT
GPT2-large	0.970	0.989	0.113	$3.41e-8$*
GPT2-XL	0.972	0.958	0.0286*	$8.14e-6$*
GPT-J	0.865	0.770	0.317	0.976

Table 4: Comparative analysis of efficacy in HIERARCHYDATA: t-test results for specific vs. abstract level distributions ($*$ indicates corresponding entry has p-value below 0.05).

the efficacy across the two hierarchical levels. Figure 5 shows one of the examples. Furthermore, we conduct t-tests on two independent samples to determine whether the mean of the specific level distribution is significantly lower than that of the abstract level distribution. The p-values obtained are documented in Table 4. This finding underscores a markedly lower efficacy in editing knowledge at higher hierarchical levels (more abstract knowledge). Significantly, this discrepancy indicates that hierarchical relationships profoundly affect the efficacy of specific editing techniques, like ROME and MEMIT, when applied to particular models, such as GPT2-Large and GPT2-XL. For fine-tuning and LoRA, the results do not appear to be significant, possibly because these methods can address knowledge at different hierarchical levels similarly. But, how about GPT-J?

GPT-J can understand perplexing knowledge better From the previous experiment, we observe that GPT-J did not show any difference in edit efficacy when editing higher hierarchy and lower hierarchy knowledge. To determine if GPT-J finds the same knowledge less perplexing compared to GPT-2L and GPT-2XL, we generated a heatmap of each knowledge's 'perplexingness' in the HIERARCHYDATA for each model, as shown in Figure 4. Each line represents a piece of knowledge in the HIERARCHYDATA, sorted by "perplexingness" in the GPT-2L model. We observed that GPT-J appears darker in the heatmap, indicating it finds the same knowledge less perplexing.

To assess the statistical significance of this observation, we conduct paired t-tests comparing the perplexingness values of GPT-J to those of GPT-2L and GPT-2XL. The resulting p-values were $5.71e-9$ and $6.84e-7$, respectively, indicating a very significant difference. This suggests that GPT-J indeed finds the same knowledge less perplexing than GPT-2L and GPT-2XL, implying that GPT-J is more receptive to learning new things. Additionally, this means GPT-J can learn more beyond hierarchical relationships, and various factors will influence its edit efficacy.

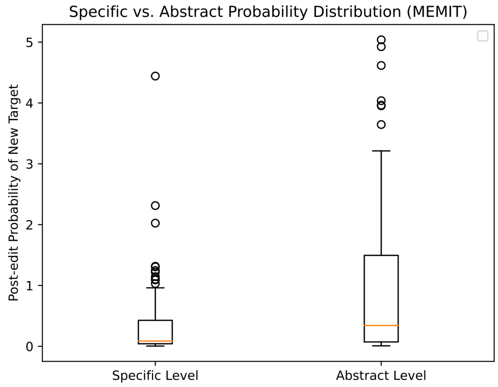


Figure 5: The post-edit probability (lower probability means higher edit efficacy) of editing GPT2-XL with MEMIT on specific vs. abstract knowledge in the HIERARCHYDATA.

Lack of Significant Findings Across Knowledge Categories Besides hierarchical relations, we also try to find if categories of knowledge would affect "perplexingness". We attempt to categorize the data based on the types of knowledge; however, this method does not yield any significant insights related to "perplexingness".

6 Discussion

Do different models have different mechanisms of saving perplexing knowledge? Our experimental results reveal intriguing variations in how different models handle perplexing knowledge, particularly in the context of editing. Specifically, the application of ROME and MEMIT to GPT-J exhibits a notably low Pearson correlation between "perplexingness" and editing efficacy. Moreover, within the HIERARCHYDATA context, these correlations appear insignificant. Additionally, the influence of hierarchical relations on the editing efficacy of ROME and MEMIT when applied to GPT-J seems negligible. This suggests that GPT-J may employ a unique mechanism for storing and processing different hierarchy level knowledge compared to other models. These differences highlight the need to comprehend each model's unique architecture and methods for handling perplexing concepts, suggesting a move towards tailored editing strategies.

Why should more abstract knowledge be harder to edit? An intuition is that when editing towards a hypernym ("animal" → "plant"), it is assumed that the hyponym ("cat" → "plant") is edited as well, making the edit of hypernym inherently harder. Yet, the dependent knowledge is usually not edited, for popular editing methods (Li et al., 2023).

Are there other factors that may influence the "perplexingness"? The investigation into the responsiveness of different editing techniques to perplexing knowledge reveals that FT and LoRA are seemingly unaffected by the hierarchical structure of knowledge. Notably, there exists a pronounced correlation between "perplexingness" and the efficacy of edits. This suggests that while FT and LoRA are adept at navigating the hierarchical relationships among words, they falter when addressing the inherent "perplexingness" present within the knowledge. This observation leads to the hypothesis that additional factors, beyond hierarchical complexity, play a pivotal role in influencing "perplexingness" when employing FT and LoRA for knowledge editing.

More understanding of model editing The impact of "perplexingness" on the efficacy of various editing methodologies can vary significantly. Moreover, the manner in which different models interpret, process, and encode the "perplexingness" of knowledge also differs. This suggests a complex interplay between the editing methods used and the intrinsic mechanisms of the models, underscoring the need for a nuanced understanding of both to optimize knowledge editing strategies.

Recommendations to future model editors a. Future model editing efforts should pay attention to understanding the nature of the knowledge being edited, particularly its level of "perplexingness". To aid in this endeavor, we have introduced a hierarchy dataset designed to facilitate it. It is crucial to ensure that editing methods are versatile and effective across a diverse range of data types. b. Moreover, adopting different editing approaches tailored to the specificities of each model can significantly enhance the success of edits. And when edit hierarchy knowledge, we can try to use edit methods like fine-tune or LoRA. It may dismiss the influence of hierarchy data. c. Also, we should pay attention to the side effect of knowledge edit.

Limitation a. In this paper, we focus on a short hierarchy chain to facilitate the comparison between higher and lower hierarchy levels. We have not yet explored longer hierarchy chains. b. The experiment can be scaled up, including the use of larger models and larger datasets. c. Additional types of evaluation can be applied. For instance, we could ask language models specific questions to determine if the knowledge has actually been edited. However, this approach is very labor-intensive and was not implemented in this study.

7 Conclusion

In our study, we focus on the challenges of updating large language models (LLMs) with perplexing knowledge. We meticulously define "perplexingness" and efficacy respectively. Through a comprehensive analysis using the COUNTERFACT dataset, we identify a significant negative correlation between the "perplexingness" of the new knowledge and the efficacy of the edits across diverse scenarios. This core finding underscores the variable efficacy of editing methods in handling knowledge with different levels of initial "perplexingness".

Furthermore, we develop a specialized dataset HIERARCHYDATA, consisting of hyponym-hypernym pairs. This dataset, emphasizing hierarchical relations, serves as a tool for a more contextual evaluation of edit efficacy. We undertake a thorough review of current knowledge editing methodologies using this dataset. Our findings reveal that abstract knowledge are inherently more perplexing to LLMs than their specific counterparts. Also, our investigation into the impact of hierarchical knowledge structures on edit outcomes reveal that more abstract knowledge exhibits lower editing efficacy in some scenarios. Our methodology and dataset collectively provide a novel and rigorous approach to evaluating the efficacy of knowledge edits, offering valuable insights into the factors that contribute to their success or failure.

Our investigation into the targeted editing of knowledge within LLMs sheds light on a previously underexplored facet of model editing technology. The findings underscore the challenges associated with editing knowledge that spans various levels of "perplexingness", revealing significant discrepancies in editing efficacy. This research not only enriches our understanding of the inherent complexities in model editing but also sets a foundational basis for the development of more sophisticated editing methodologies in the future. By pushing the boundaries of our current capabilities, we move closer to achieving more refined and precise manipulations of knowledge within these advanced AI systems, marking a significant step forward in the evolution of LLMs.

References

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*.

Anshita Gupta, Debanjan Mondal, Akshay Sheshadri, Wenlong Zhao, Xiang Li, Sarah Wiegreffe, and Niket Tandon. 2023. Editing common sense in transformers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8214–8232.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2024. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *Advances in Neural Information Processing Systems*, 36.

Jason Hoelscher-Obermaier, Julia Persson, Esben Kran, Ioannis Konstas, and Fazl Barez. 2023. Detecting edit failures in large language models: An improved specificity benchmark. *arXiv preprint arXiv:2305.17553*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Zichao Li, Ines Arous, Siva Reddy, and Jackie Chi Kit Cheung. 2023. Evaluating dependencies in fact editing for language models: Specificity and implication awareness. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7623–7636.

Xinbei Ma, Tianjie Ju, Jiyang Qiu, Zhuosheng Zhang, Hai Zhao, Lifeng Liu, and Yulong Wang. 2024. Is it possible to edit large language models robustly? *arXiv preprint arXiv:2402.05827*.

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

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

Jingcheng Niu, Andrew Liu, Zining Zhu, and Gerald Penn. 2024. What does the Knowledge Neuron Thesis Have to do with Knowledge? In *ICLR*.

Yasumasa Onoe, Michael JQ Zhang, Shankar Padmanabhan, Greg Durrett, and Eunsol Choi. 2023. Can LMs learn new entities from descriptions? challenges in propagating injected knowledge. *arXiv preprint arXiv:2305.01651*.

Domenic Rosati, Robie Gonzales, Jinkun Chen, Xuemin Yu, Melis Erkan, Yahya Kayani, Satya Deepika Chavatapalli, Frank Rudzicz, and Hassan Sajjad. 2024. Long-form evaluation of model editing. *arXiv preprint arXiv:2402.09394*.

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*.

Suhang Wu, Minlong Peng, Yue Chen, Jinsong Su, and Mingming Sun. 2023. Eva-kellm: A new benchmark for evaluating knowledge editing of llms. *arXiv preprint arXiv:2308.09954*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing Knowledge Editing in Language Models via Multi-Hop Questions. *arXiv preprint arXiv:2305.14795*.

A Correlation of "perplexingness" and efficacy in COUNTERFACT

We plot the "perplexingness" (pre-edit probabilities of the new target) against the efficacy (post-edit probabilities of the new target) to visually analyze their relationship. This analysis is conducted using the first 2000 groupings from the COUNTERFACT dataset. Figure 6 displays the scatter plot for editing methods applied to GPT2-Large. Similarly, Figure 7 presents the scatter plot for methods used on GPT2-XL, and Figure 8 illustrates the scatter plot for edits performed on GPT-J(6B).

B Correlation of "perplexingness" and efficacy in HIERARCHYDATA

To visually explore the relationship between "perplexingness" and editing efficacy, we plot these dimensions against each other using 198 groupings from the HIERARCHYDATA dataset. Figure 9 shows the scatter plot highlighting the effects of editing methods on the GPT2-Large model. Likewise, Figure 10 demonstrates the scatter plot for the GPT2-XL model, and Figure 11 displays the scatter plot for edits on the GPT-J(6B) model, providing a clear visual representation of how "perplexingness" correlates with the efficacy of knowledge edits across different models.

C Specific vs. Abstract Probability Distribution in HIERARCHYDATA

We conduct a comparative analysis by plotting the efficacy distributions for data at both specific and abstract hierarchical levels, utilizing 198 groupings from the HIERARCHYDATA dataset—comprising an equal split of 99 specific-level instances and 99 abstract-level instances. Figure 12 showcases the box plot for editing methods applied to the GPT2-Large model. In a similar vein, Figure 13 displays the box plot for techniques employed on the GPT2-XL model, while Figure 14 reveals the box plot corresponding to edits made on the GPT-J(6B) model.

D Pre-edit Specific vs. Abstract Probability Distribution in HIERARCHYDATA

We perform a comparative analysis of the "perplexingness" across both specific and abstract hierarchical levels by plotting their distributions. This analysis is based on 198 instances from the HIERARCHYDATA dataset, evenly divided between 99 specific-level and 99 abstract-level cases. Figure 15 presents the box plots, illustrating the impact of editing methods on the GPT2-Large, GPT2-XL, and GPT-J(6B) models, thereby offering insights into the variation of "perplexingness" across different levels of hierarchy and models.

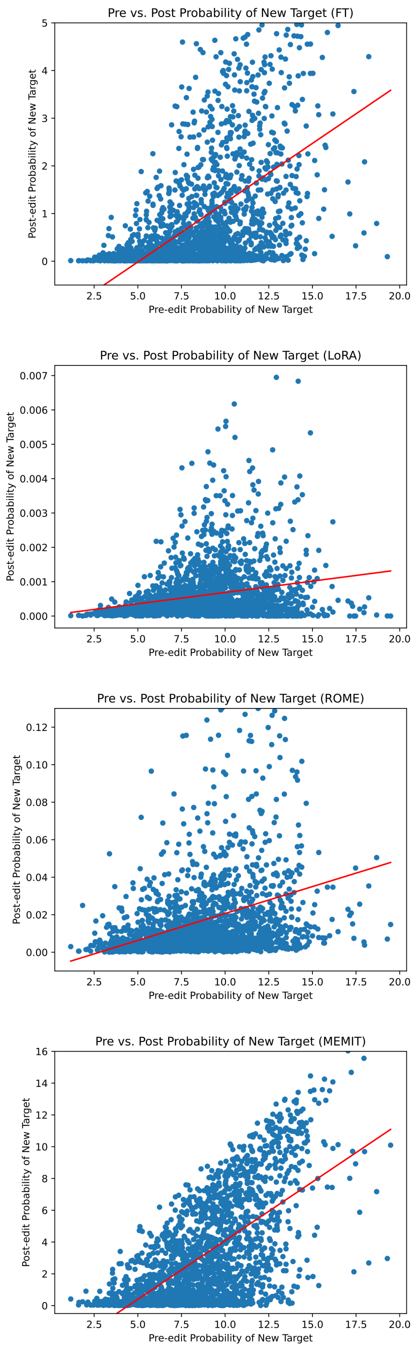


Figure 6: Pre vs. post probability of new knowledge (COUNTERFACT) on GPT2-Large using a. FT (upper left) b. LoRA (upper right) c. ROME (lower left) d. MEMIT (lower right).

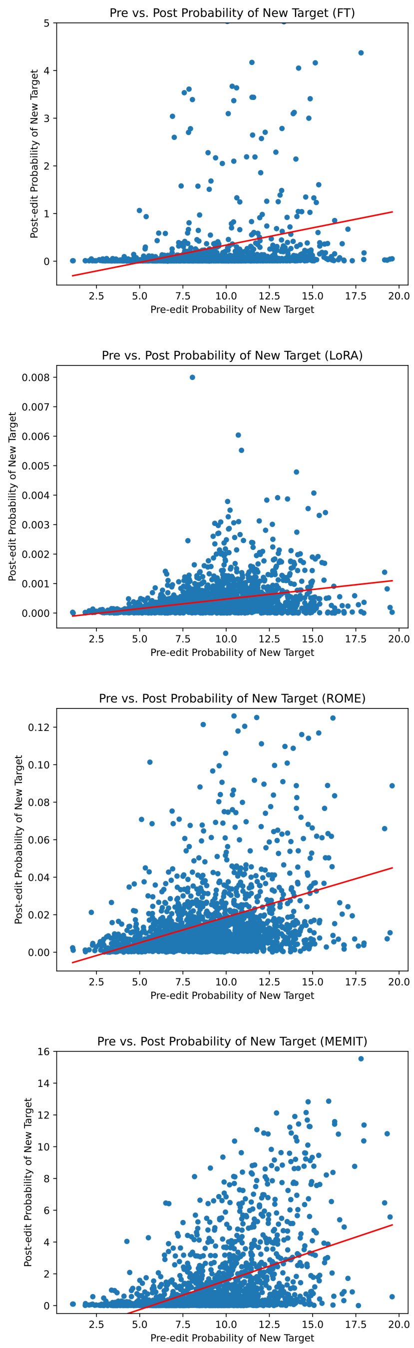
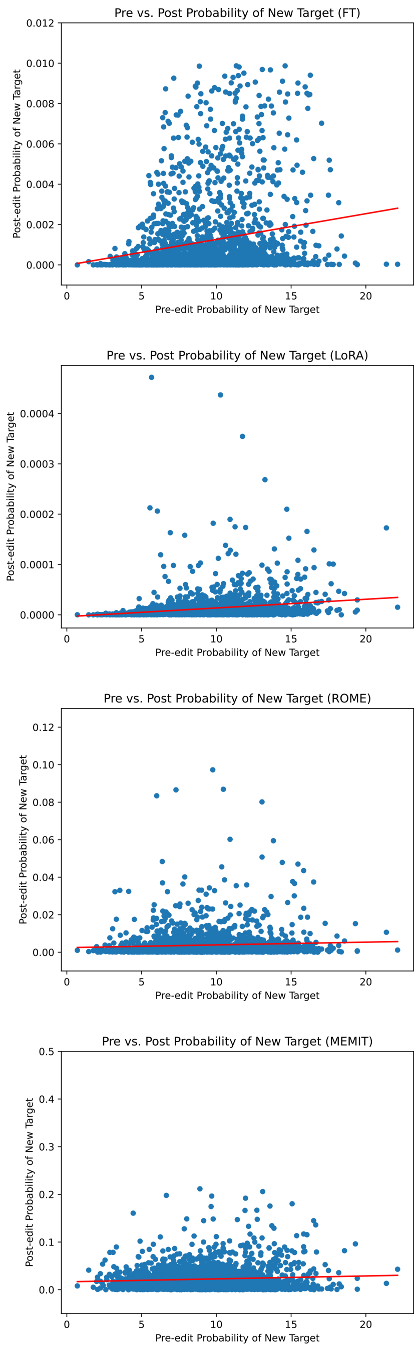


Figure 7: Pre vs. post probability of new knowledge (COUNTERFACT) on GPT2-XL using a. FT (upper left) b. LoRA (upper right) c. ROME (lower left) d. MEMIT (lower right).



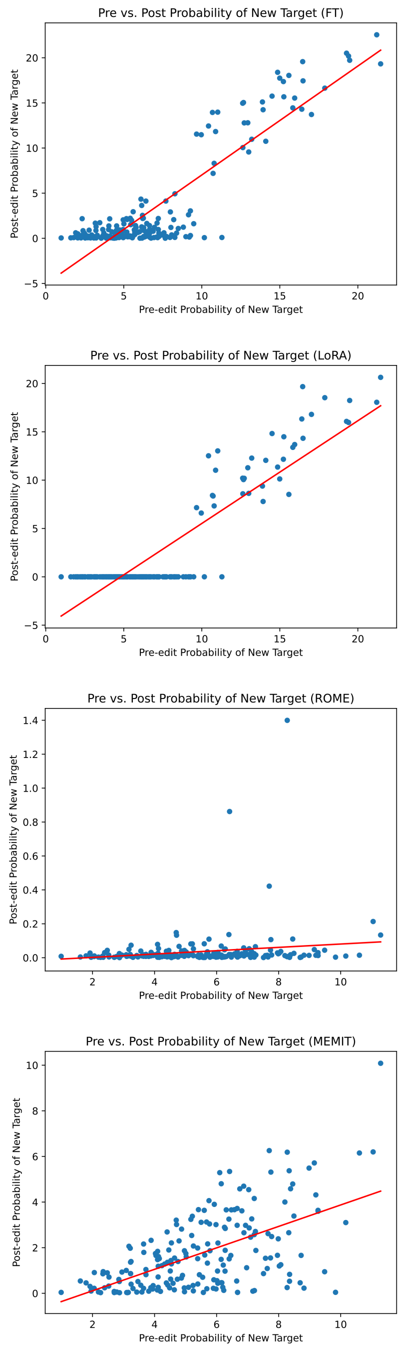


Figure 8: Pre vs. post probability of new knowledge (COUNTERFACT) on GPT-J(6B) using a. FT (upper left) b. LoRA (upper right) c. ROME (lower left) d. MEMIT (lower right).

Figure 9: Pre vs. post probability of new knowledge (HIERARCHYDATA) on GPT2-Large using a. FT (upper left) b. LoRA (upper right) c. ROME (lower left) d. MEMIT (lower right).

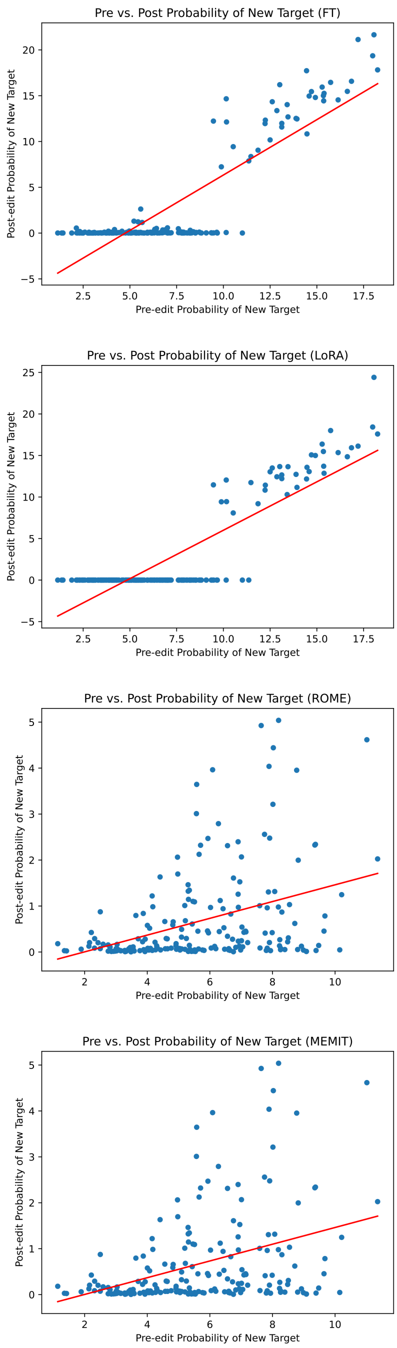


Figure 10: Pre vs. post probability of new knowledge (HIERARCHYDATA) on GPT2-XL using a. FT (upper left) b. LoRA (upper right) c. ROME (lower left) d. MEMIT (lower right).

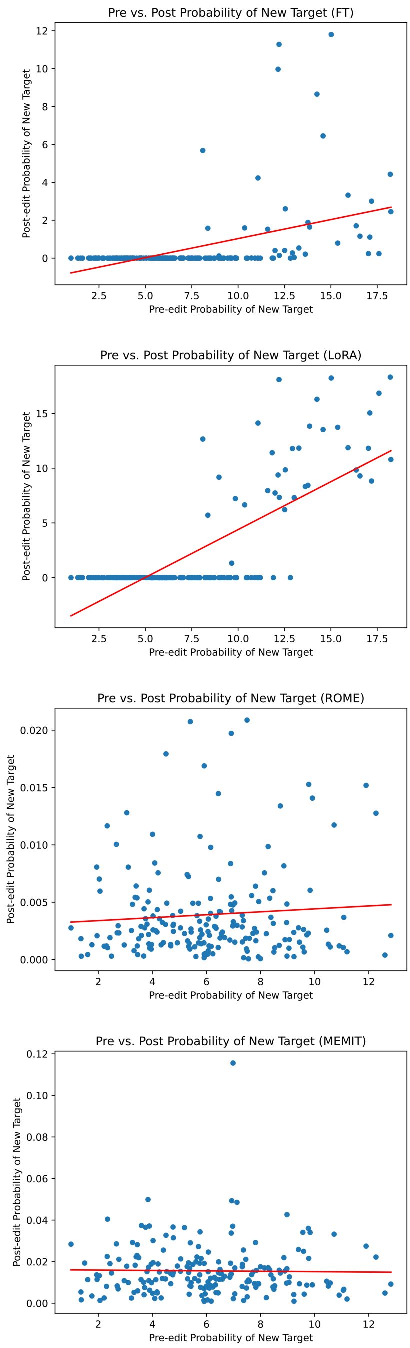


Figure 11: Pre vs. post probability of new knowledge (HIERARCHYDATA) on GPT-J(6B) using a. FT (upper left) b. LoRA (upper right) c. ROME (lower left) d. MEMIT (lower right).

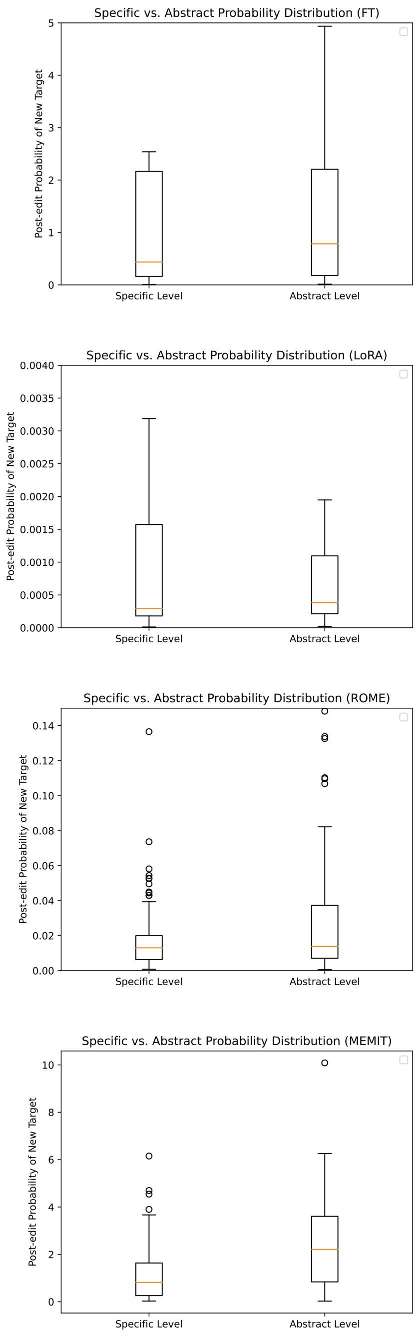


Figure 12: Specific vs. abstract probability distribution (HIERARCHYDATA) on GPT2-Large using a. FT (upper left) b. LoRA (upper right) c. ROME (lower left) d. MEMIT (lower right).

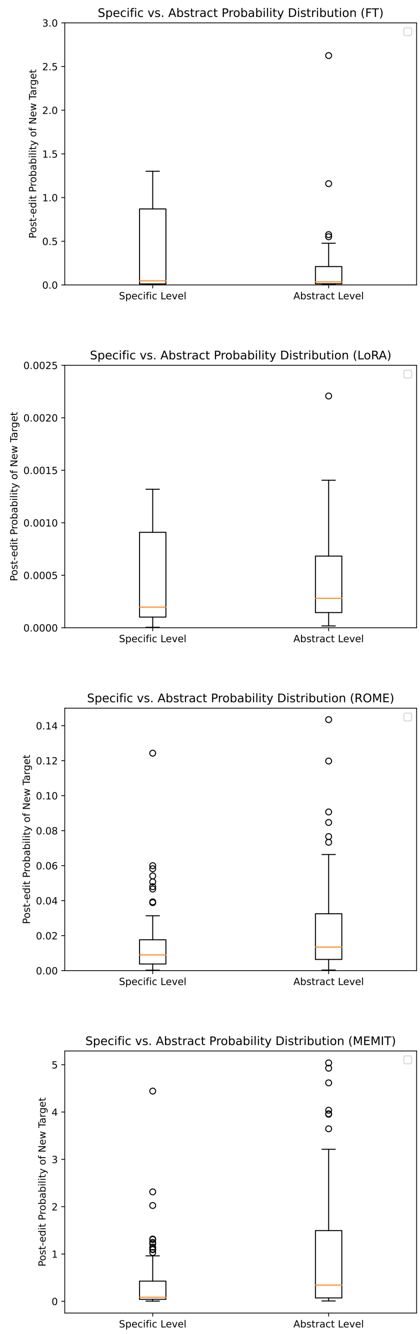


Figure 13: Specific vs. abstract probability distribution (HIERARCHYDATA) on GPT2-XL using a. FT (upper left) b. LoRA (upper right) c. ROME (lower left) d. MEMIT (lower right).

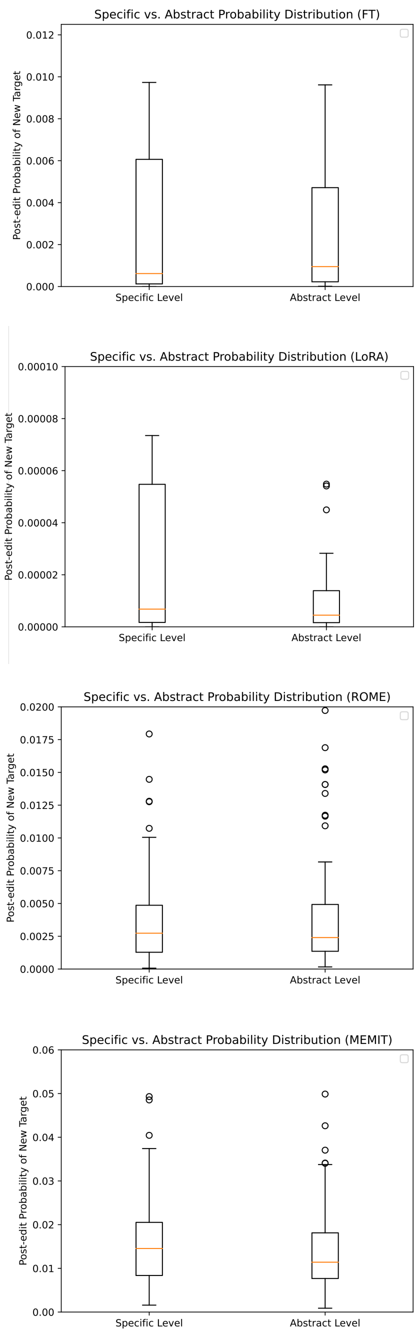


Figure 14: Specific vs. abstract probability distribution (HIERARCHYDATA) on GPT-J(6B) using a. FT (upper left) b. LoRA (upper right) c. ROME (lower left) d. MEMIT (lower right).

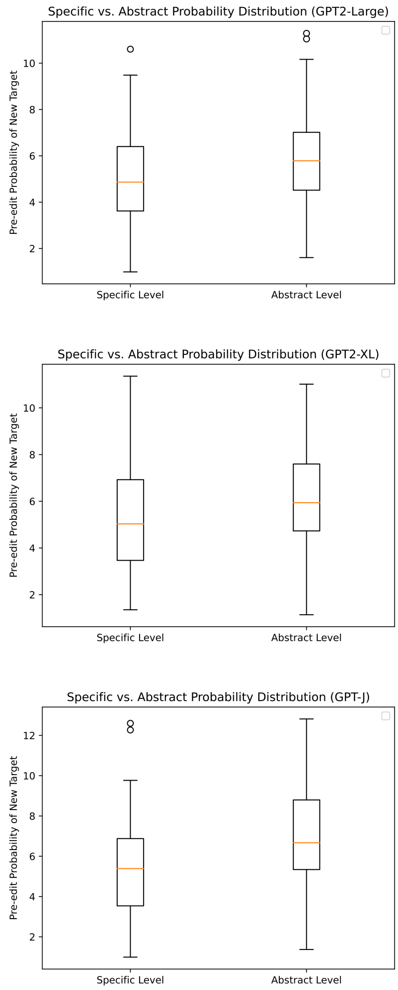


Figure 15: Pre-edit specific vs. abstract probability distribution (HIERARCHYDATA) on a. GPT2-Large (upper left) b. GPT2-XL (upper right) c. GPT-J(6B) (below).