

Model Editing Harms General Abilities of Large Language Models: Regularization to the Rescue

Jia-Chen Gu^{1*}, Hao-Xiang Xu^{2*}, Jun-Yu Ma², Pan Lu¹,
Zhen-Hua Ling², Kai-Wei Chang¹, Nanyun Peng¹

¹University of California, Los Angeles

²University of Science and Technology of China

{gujc, panbruin}@ucla.edu, {nh2001620, mgy1999}@mail.ustc.edu.cn
zhling@ustc.edu.cn, {kwchang, violetpeng}@cs.ucla.edu

Abstract

Model editing is a technique that edits the large language models (LLMs) with updated knowledge to alleviate hallucinations without resource-intensive retraining. While current model editing methods can effectively modify a model’s behavior within a specific area of interest, they often overlook the potential unintended side effects on the general abilities of LLMs such as reasoning, natural language inference, and question answering. In this paper, we raise concerns that model editing’s improvements on factuality may come at the cost of a significant degradation of the model’s general abilities. We systematically analyze the side effects by evaluating four popular editing methods on three LLMs across eight representative tasks. Our extensive empirical experiments show that it is challenging for current editing methods to simultaneously improve factuality of LLMs and maintain their general abilities. Our analysis reveals that the side effects are caused by model editing altering the original model weights excessively, leading to *overfitting* to the edited facts. To mitigate this, a method named **RECT** is proposed to regularize the edit update weights by imposing constraints on their complexity based on the **RE**lative Change in weightT. Evaluation results show that **RECT** can significantly mitigate the side effects of editing while still maintaining over 94% editing performance¹.

1 Introduction

As real-world knowledge is dynamically increasing and updating, existing large language models (LLMs) need to constantly incorporate the inherit knowledge and up-to-date information for life-long learning. Despite continual training, LLMs inevitably manifest hallucinations caused by missing, false or outdated knowledge embedded in

* Equal contribution.

¹<https://github.com/JasonForJoy/Model-Editing-Hurt>

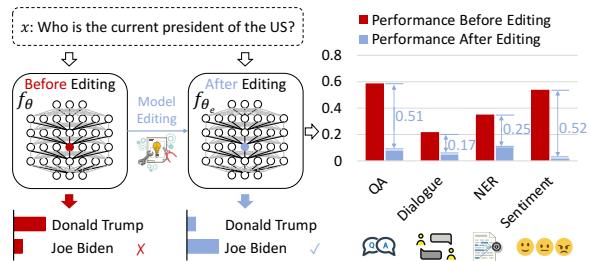


Figure 1: Demonstration of model editing and its impact on the general abilities of LLMs. Although the factuality of the model has been improved, the general abilities of LLMs, such as question answering, dialogue, named entity recognition, sentiment analysis, have been substantially impaired after editing. f_θ / f_{θ_e} denotes the models before / after editing.

their parameters (Zhang et al., 2023; Peng et al., 2023; Ji et al., 2023). Due to the intensive computational cost of retraining LLMs, researchers have increasingly focused on *model editing* (a.k.a., *knowledge editing*) (Sinitzin et al., 2020; Cao et al., 2021; Dai et al., 2022; Mitchell et al., 2022b; Meng et al., 2022, 2023; Yao et al., 2023; Zhong et al., 2023; Ma et al., 2023; Zhang et al., 2024). This task is to efficiently modify a model’s behavior within a specific area of interest through targeted interventions without resource-intensive model retraining.

At present, the assessment of editing methods typically involves evaluation along three critical dimensions (Yao et al., 2023). First, *reliability* ensures the edited model can accurately recall the specific edited fact. Second, *generalization* validates the adaptability of the edited model by assessing the model’s ability to recall the fact under diverse paraphrase prompts. Finally, *locality* checks if the edited model’s output for unrelated inputs remains consistent after editing. These multi-faceted criteria collectively contribute to a nuanced understanding of the effectiveness and robustness of editing methods.

In this paper, we put forward a critical concern regarding the overall robustness and adaptability of edited models. As shown in Figure 1, while model editing methods have demonstrated improved factuality, it may come at the significant cost of the general abilities of LLMs such as summarization, question answering (QA), natural language inference. We argue that improving model factuality must be balanced with the need to maintain effectiveness across a range of abilities.

In light of the above issues, we systematically study if model editing hurts the general abilities of LLMs. This work studies model editing in the *single*- versus *sequential*-editing and *instance*-versus *batch*-editing settings. The edited models are evaluated on a variety of downstream tasks to see if there are any side effects on performance *before* and *after* editing. Extensive empirical experiments are conducted on **four popular editing methods**: KN (Dai et al., 2022), MEND (Mitchell et al., 2022a), ROME (Meng et al., 2022), and MEMIT (Meng et al., 2023) applied to **three representative LLMs**: GPT-2 XL (1.5B) (Radford et al., 2019), LLaMA-1 (7B) (Touvron et al., 2023a), and LLaMA-2 (7B) (Touvron et al., 2023b). **Eight representative tasks** including reasoning (Cobbe et al., 2021), natural language inference (Dagan et al., 2005), open-domain QA (Kwiatkowski et al., 2019), closed-domain QA (Clark et al., 2019), dialogue (Cui et al., 2020), summarization (Gliwa et al., 2019), named entity recognition (Sang and Meulder, 2003), and sentiment analysis (Socher et al., 2013) are employed to understand the impact of model editing on the general abilities of LLMs.

Experimental results show that existing LLMs are not robust to weight perturbations, and editing even a few parameters can significantly affect their general abilities. Strikingly, with a single pass of editing involving less than 1% parameters, LLaMA-1 (7B) exhibited a drastic performance degradation to nearly 0 on all the tasks we tried. These results demonstrate that current editing algorithms struggle to work effectively in tandem with LLMs to simultaneously improve model factuality and maintain general abilities.

Furthermore, our analysis of the causes of side effects reveals that current model editing methods change the original model weights too much, resulting in *overfitting* to new editing facts. The accumulation of overfitting across multiple edits can amplify the negative impact on the general abilities of LLMs. As a result, the edited model can

recall new editing facts well but fails to generalize to various downstream tasks. To this end, we design a regularization method named **RECT** (**R**elative **C**hange in **weigT**) to prevent overfitting. Basically, this regularization discourages overly complex editing updates that are more likely to overfit. Specifically, the top- $k\%$ elements in an edit update weight that change the most according to relative change in weights are considered as the principal editing information and keep their original values. While for the remaining elements in this edit update weight, they are treated as minor contributions to editing and set to zero for regularization. Evaluation results show that the edited models regularized by RECT can effectively mitigate the side effects of editing while still maintaining over 94% editing performance

In summary, we demonstrate that although model editing is effective in updating parametric knowledge in a resource-efficient and target-specific way, current methods still have significant flaws in preserving the general abilities of LLMs. Existing research on model editing excessively pursued altering a model’s behavior under specific knowledge while overlooked the premise of not compromising general abilities. This paper points out the urgent shortcomings in model editing and proposes a regularization method to prevent overfitting across multiple edits to rescue the general abilities, calling for follow-up research efforts on trustworthy and robust model editing.

2 Related Work

Many studies have investigated model editing, including memory-based, meta-learning, and locate-then-edit (Wang et al., 2024a; Yao et al., 2023). Memory-based methods do not modify model weights but store the editing facts with an external memory (Mitchell et al., 2022b; Zhong et al., 2023). Mitchell et al. (2022b) stored edits in a base model and learned to reason over them to adjust its predictions as needed. The latter two classes of methods are developed to directly modify the internal parameters of models, which is the focus of this paper. On the one hand, meta-learning methods train a hypernetwork to get gradient changes to update model parameters (Cao et al., 2021; Mitchell et al., 2022a). Cao et al. (2021) utilized a hypernetwork to predict parameter shift at test time. Mitchell et al. (2022a) learned to transform the fine-tuning gradient into a low-rank decomposition of

the gradient. On the other hand, locate-then-edit methods first locate knowledge neurons in LLMs that exhibit a positive correlation with a knowledge expression, and then modify them accordingly (Dai et al., 2022; Meng et al., 2022, 2023). In particular, Dai et al. (2022) computed the contribution of each neurons to a certain knowledge, then updated or erased knowledge by modifying these neurons with the embedding vectors of facts. Meng et al. (2022) located multi-layer perceptron (MLP) storing factual knowledge, and then edited such knowledge by injecting new key-value pair in the MLP module. Besides, some works investigate the evaluation paradigm for model editing (Zhong et al., 2023; Cohen et al., 2024; Ma et al., 2023; Li et al., 2024; Hase et al., 2023; Wu et al., 2023a; Gandikota et al., 2023; Wang et al., 2024b). For example, Cohen et al. (2024) introduced the ripple effects of editing, suggesting that editing a particular fact implies that many other facts need to be updated. Additionally, recent works have also applied editing in various domains, such as changing model personality (Mao et al., 2023), editing multimodal models (Cheng et al., 2023), protecting users privacy (Wu et al., 2023b), etc.

A main difference between this work and previous related studies should be highlighted. These approaches target at designing editing algorithms to improve or evaluation paradigms to assess the editing performance. In contrast, this study rethinks model editing and explores if current editing methods inadvertently cause the potential side effects on the underlying general abilities of LLMs. The contemporaneous work (Gupta et al., 2024) presents a similar finding that model editing at scale leads to catastrophic forgetting, but no mitigation method is proposed. To the best of our knowledge, this paper makes the first call for attention to side effects on a variety of tasks beyond editing performance by presenting a systematical evaluation of four editing methods on three LLMs covering eight tasks. Besides, we also analyze the causes of side effects, and propose a regularization method to prevent editing overfitting.

3 Preliminary

Model editing involves modifying the memorized facts contained in LMs without retraining to better suit specific tasks or requirements. Various kinds of complex learned beliefs such as logical, spatial, or numerical knowledge are expected to be edited.

In this paper, we study editing factual knowledge in the form of (subject s , relation r , object o), e.g., ($s = \text{United States}$, $r = \text{President of}$, $o = \text{Donald Trump}$). An LM is expected to recall a memory and predict the next token(s) representing o given a natural language prompt $p(s, r)$ such as “*The President of the United States is*”. Editing a fact is to insert a new knowledge triple (s, r, o^*) in place of the current one (s, r, o) , where these two triples share the same subject and relation. An editing operation is represented as $e = (s, r, o, o^*)$ for brevity. Given a set of editing facts $\mathcal{E} = \{e_1, e_2, \dots\}$ and a model f , model editing involves learning a function K that yields an edited LM $f^* : K(f, \mathcal{E}) = f^*$.

To evaluate the effectiveness of editing methods, previous works focus on evaluation along three dimensions (Cao et al., 2021; Mitchell et al., 2022a; Meng et al., 2022, 2023). First and foremost is *reliability*, aiming to ascertain the ability of the edited model to accurately recall the specific editing facts. The second dimension *generalization* seeks to validate the adaptability of the edited model by assessing its ability to recall the editing facts under diverse paraphrase prompts. The last dimension *locality* (a.k.a., *specificity*) is employed to verify the stability of the edited model by examining whether its output for unrelated inputs remains consistent after editing. Due to limited space, readers can refer to Appendix A for more detailed explanations and examples of the evaluation metrics.

4 Analysis of Side Effects of Editing

4.1 Evaluation Paradigm

This paper systematically studies the side effects of model editing in the *single*- versus *sequential*-editing and *instance*- versus *batch*-editing settings. Figure 2 illustrates these experimental settings. The edited models are evaluated under the zero-shot setting on a variety of downstream tasks unrelated to editing facts to understand the performance *before* and *after* editing.

Single- vs. Sequential-editing Single-editing involves examining the reliability and impact of making a single editing operation to a model. Specifically, it focuses on understanding how a model adapts to such a single alteration, and the implicit effect of such specific modifications on the overall performance. It is worth noting that a single editing operation can contain either only one editing instance or multiple ones in a batch, which

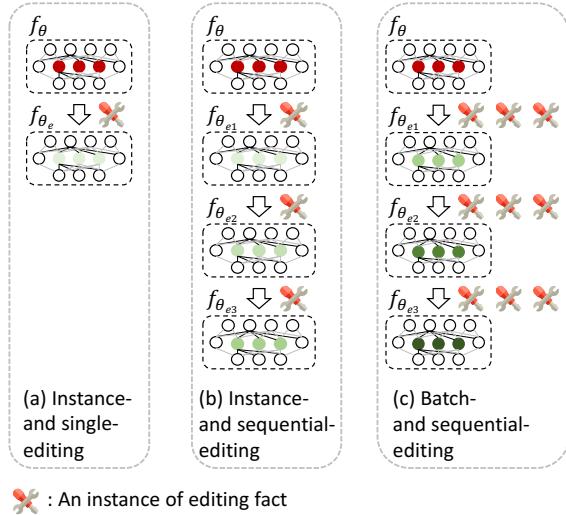


Figure 2: Illustration of the settings of (a) *single- and instance-editing*, (b) *sequential- and instance-editing*, and (c) *sequential- and batch-editing*. The darker units correspond to more edits.

is further discussed later. In practice, there are often situations where only a particular change is needed, so it’s crucial to understand how effectively the model integrates and preserves that individual edit. Therefore, evaluating the robustness to a single edit is crucial in determining its ability to retain the intended changes and overall performance.

In contrast to single-editing, multiple editing operations are conducted successively in sequential-editing (Huang et al., 2023). Similarly, each editing operation in sequential-editing can also contain either only one editing instance or multiple ones in a batch. Ideally, models should retain the changes from previous edits when carrying out a new one (Yao et al., 2023), which is decisive for the continual learning of future LLMs. Therefore, whether edited models can still maintain its general abilities after sequential editing is one of the important characteristics that should be considered. For this analysis, how the performance of edited models on a variety of tasks changes as the number of edits increases will be explored.

Instance- vs. Batch-editing Instance-editing refers to using only one instance per editing operation to make specific and targeted adjustments to individual pieces of knowledge within LLMs, regardless of the single- or sequential-editing settings. This setting is particularly valuable in situations where certain instances present unique challenges or outliers that require specialized treatment. These fine-grained alterations to model behaviors over individual instances are expected to

contribute to more adaptable and accurate LLMs.

The real world is ever-changing, so there is a huge amount of knowledge that needs to be dynamically added and updated into LLMs. Despite the effectiveness of many instance-editing methods (Dai et al., 2022; Meng et al., 2022; Ma et al., 2023), ultimately at most a few dozens of pieces of knowledge can be updated (Mitchell et al., 2022b), due to their relatively low but still non-negligible editing cost for a single instance. Since naive sequential applications of current state-of-the-art model editing methods fail to scale up (Meng et al., 2023), one may wish to update hundreds or thousands of facts simultaneously in batch-editing. Notably, batch-editing can also be coupled with both the single- or sequential-editing settings.

Zero-shot Learning Zero-shot learning aims to solve tasks without labeled training examples, and recent studies have demonstrated the superiority of LLMs for zero-shot learning (Brown et al., 2020; Wei et al., 2022; Chowdhery et al., 2023). Following these studies, we explore the zero-shot performance of unedited and edited models on a variety of tasks. Given a task instruction and a test problem that are concatenated as the input, the model is expected to generate a target text to address this problem. The instructions and input formats of different tasks are shown in Appendix B, which are taken from or inspired by Qin et al. (2023).

4.2 Evaluation Setup

We briefly introduced the experimental setup regarding editing methods, editing datasets, selected LLMs, and representative tasks here. Readers can refer to their corresponding papers for more details.

Editing Methods Four popular editing methods were selected: (1) *KN* (Dai et al., 2022) involved identifying neurons linked to knowledge expression using gradient-based attributions and then enhancing the MLP layer by adding scaled embedding vectors to those specific neurons. (2) *MEND* (Mitchell et al., 2022a) learned a hypernetwork to produce weight updates by decomposing the fine-tuning gradients into rank-1 form. (3) *ROME* (Meng et al., 2022) involved localizing factual knowledge in a specific Transformer MLP layer and then updating it by directly writing new key-value pairs into the MLP module. (4) *MEMIT* (Meng et al., 2023) expanded the capabilities of ROME by enabling the editing of large

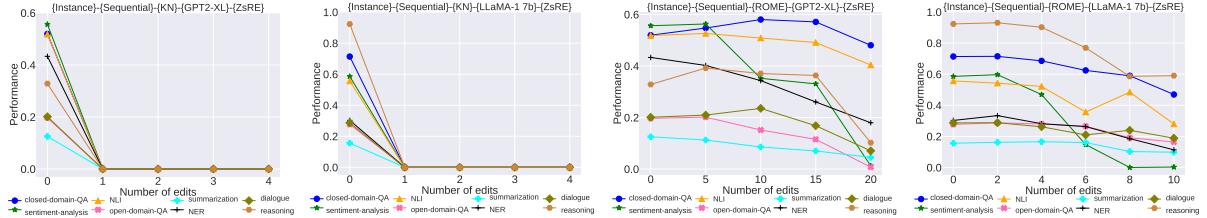


Figure 3: Performance on general tasks of edited models using KN or ROME to edit GPT-2 XL or LLaMA-1 (7B) as the number of edits increases in *instance- and sequential-editing*.

amounts of factual data through the updating of a sequence of MLP layers. It is notable that only MEND and MEMIT support batch-editing. All experiments were conducted using the EasyEdit tool (Wang et al., 2024a), ensuring standardized and reproducible evaluation. All editing instances were randomly sampled from the editing dataset. Readers can refer to Appendix C for details of these editing methods.

Editing Dataset The popular model editing dataset Zero-Shot Relation Extraction (ZsRE) (Levy et al., 2017) used in previous work (Cao et al., 2021; Meng et al., 2022; Yao et al., 2023) was adopted in our experiments. ZsRE is a QA dataset using question rephrasings generated by back-translation as the equivalence neighborhood. Each input is a question about an entity, and plausible alternative edit labels are sampled from the top-ranked predictions of a BART-base model trained on ZsRE.

Selected LLMs Experiments were conducted on three LLMs: GPT-2 XL (1.5B) (Radford et al., 2019), LLaMA-1 (7B) (Touvron et al., 2023a), and LLaMA-2 (7B) (Touvron et al., 2023b).

Downstream Tasks and Metrics To extensively explore whether model editing has side effects on the general abilities of LLMs, eight representative tasks were adopted: (1) *Reasoning* on the GSM8K (Cobbe et al., 2021), and the results were measured by solve rate. (2) *Natural language inference (NLI)* on the RTE (Dagan et al., 2005), and the results were measured by accuracy of two-way classification. (3) *Open-domain QA* on the Natural Question (Kwiatkowski et al., 2019), and the results were measured by exact match (EM) with the reference answer after minor normalization as in Chen et al. (2017) and Lee et al. (2019). (4) *Closed-domain QA* on the BoolQ (Clark et al., 2019), and the results were also measured by EM. (5) *Dialogue* on the MuTual (Cui et al., 2020), and the results were measured by selecting

one best-matched response from four available candidates, denoted as *Recall₄@1* as in Lowe et al. (2015). (6) *Summarization* on the SAMSUM (Gliwa et al., 2019), and the results were measured by the average of ROUGE-1, ROUGE-2 and ROUGE-L as in Lin (2004). (7) *Named entity recognition (NER)* on the CoNLL03 (Sang and Meulder, 2003), and the results were measured by entity-level F1-score. (8) *Sentiment analysis* on the SST2 (Socher et al., 2013), and the results were measured by accuracy of two-way classification.

4.3 Results

Impact of Sequential-editing Since single-editing can be regarded as a special case of sequential-editing when the number of edits is 1, this subsection mainly discussed instance- and sequential-editing. KN and ROME that support instance-editing but not batch-editing were adopted to facilitate this exploration. MEND and MEMIT that support batch- and sequential-editing will be explored later in this subsection. Figure 3 presents the performance on general tasks of edited models using KN or ROME to edit GPT-2 XL and LLaMA-1 (7B) as the number of edits increases. Due to limited space, readers can refer to Appendix D for the results of editing LLaMA-2 (7B) which show similar trends. It can be seen that although there is only one instance per editing operation, the performance of edited models on various tasks fluctuates significantly and shows a downward trend as the number of edits increases. Strikingly, the use of KN resulted in a drastic performance degradation to nearly zero on all selected tasks with just a single edit. These findings underscore two key insights. First, the selected LLMs are not robust to weight perturbations even if less than 1% of the parameters are edited, whereby slight perturbations may significantly affect their general abilities. Second, these outcomes also shed light on the challenging nature of effectively coupling current

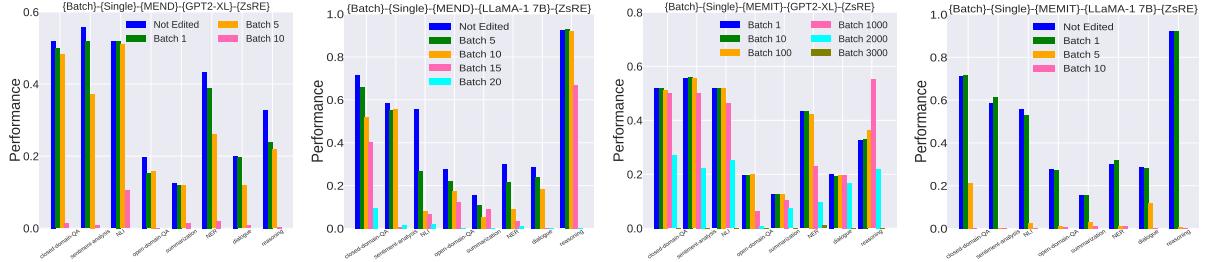


Figure 4: Performance on general tasks of edited models using MEND or MEMIT to edit GPT-2 XL or LLaMA-1 (7B) with different batch sizes in *batch- and single-editing*.

editing algorithms with LLMs. The difficulty lies in the dual objective of improving model factuality while simultaneously maintaining their general abilities. The observed trends indicate that existing editing algorithms face grand challenges in achieving this delicate balance, emphasizing the need for further research and development in the refinement of editing methodologies for LLMs.

Impact of Batch-editing This subsection delved into batch- and single-editing to explore the impact of batch size for scaling up the editing scope. Only MEND and MEMIT that supported batch-editing were adopted to facilitate this exploration. Figure 4 presents the performance on general tasks of edited models using MEND or MEMIT to edit GPT-2 XL and LLaMA-1 (7B) with different batch sizes. Readers can refer to Appendix D for the results of editing LLaMA-2 (7B). Remarkably, even with only one single editing operation, edited models exhibited a trend of performance degradation as the batch size increases in most cases. This consistent decrease in performance underlines the sensitivity of the models to increases in batch size, emphasizing the significance of carefully scaling knowledge editing for optimal updates. Therefore, we call for more research work on scalable editing to facilitate efficient editing of multiple instances.

Impact of Batch- and Sequential-editing In order to holistically take into account the interplay between batch size and sequential-editing, a joint setting of batch- and sequential-editing was explored to understand how these two factors collaboratively influence the overall performance of edited models. Figure 11 to Figure 16 in Appendix D present the performance of using MEND or MEMIT to edit GPT-2 XL, LLaMA-1 (7B) and LLaMA-2 (7B) respectively as the number of edits increases. These results also echo our observations on sequential-editing, and those on batch-editing respectively.

# edits	Manhattan dist.	% $\delta > 0.077$	% $\delta > 0.171$
1	9079.2	20.0%	10.0%
5	27072.2	49.2%	28.9%
10	52245.3	67.4%	46.2%
15	63247.5	72.8%	52.2%

Table 1: Statistics of the distinction between the final edited weight and the original unedited weight via weight change as the number of edits increases.

4.4 Analysis of Causes of Side Effects

We show that the side effects of editing come from changing the original model weights too much, resulting in *overfitting* to the editing facts. This phenomenon can be illustrated through statistics and visualization using ROME to edit GPT-2 XL.

Statistics We first show how the weights change in instance- and single-editing. Typically, one editing operation is to add an edit update weight ΔW to the original weight W , where ΔW is calculated aiming to insert new editing facts. Here, we define the absolute value of the relative change in weight $\delta = |\frac{\Delta W}{W}|$ to characterize the degree of change of each element in the update weight ΔW . The statistics show that only 20% of the elements in the update weight ΔW have δ greater than 0.077, while only 10% of the elements have δ greater than 0.171. These results are averaged by 100 random single edits in the ZsRE dataset. It can be seen that the update weight ΔW might be quite sparse, while most elements in ΔW are minor. Manhattan distance between the updated and original weights is also calculated as a measurement of distinction.

Furthermore, how the weights change in instance- and sequential-editing is shown in Table 1. As the number of edits increases, the proportion of elements whose δ is greater than a certain threshold increases significantly, and the weight is also more differentiated than the original weight. Therefore, the accumulation of overfitting

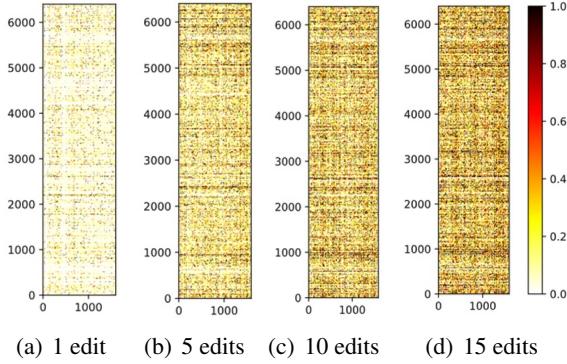


Figure 5: Visualization of the distinction between the final edited weight and the original unedited weight via weight change $|\Delta W|$ as the number of edits increases.

across multiple edits can amplify changes to the original weights.

Visualization The distinction between the final edited weight and the original unedited weight is illustrated by visualizing the weight change $|\Delta W|$ as shown in Figure 5. It reveals the consistent findings that the update weight ΔW might be quite sparse, while the accumulation across multiple edits can amplify changes to the original weights.

5 RECT: RElative Change in weightT

5.1 Approach

We have analyzed the causes of side effects in Section 4.4 that model editing changes the original model weights too much, resulting in *overfitting* to the editing facts. This type of editing overfitting occurs when a model learns to fit the new editing data too closely, capturing noise and outliers in the data rather than the underlying patterns. Furthermore, the gradual buildup of editing-induced overfitting across sequential edits can severely impair the general abilities of LLMs. Consequently, while such models may exhibit proficiency in the new editing facts, they often struggle to generalize effectively across a spectrum of downstream tasks. This phenomenon underscores the importance of mitigating overfitting during the editing process to ensure both the improvement of model factuality and the maintenance of their general abilities.

To this end, this paper designs a regularization method named **RE**lative **C**hange in **w**eig**T** (RECT) to prevent editing overfitting. Figure 6 illustrates the overview of this regularization method. Typically, one editing operation is to add an edit update weight ΔW to the original weight W

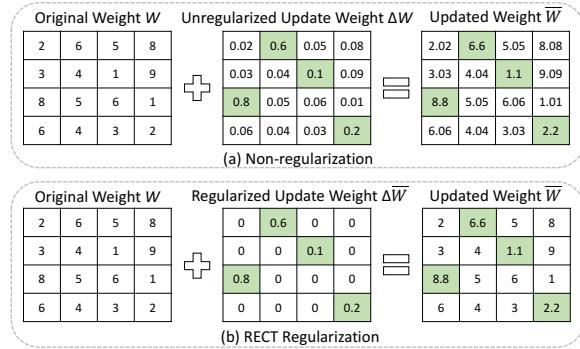


Figure 6: Comparison of (a) non-regularization and (b) the proposed RECT regularization. Elements in green denote the top- $k\%$ that change the most according to δ and are considered as the principle editing information, which should keep their original values. $k = 25$ in this figure for illustration.

to derive the updated weight \bar{W} , where ΔW is calculated aiming to insert a batch of N new editing facts $\{(s, r, o^*)_i\}_{i=1}^N$ ($N = 1$ for a single editing fact). Formally, we have:

$$\bar{W} = W + \Delta W, \quad (1)$$

$$\Delta W = f(\{(s, r, o^*)_i\}_{i=1}^N), \quad (2)$$

where function f denotes the calculation method of update weight ΔW for different editing methods, e.g., ROME (Meng et al., 2022).

Here, we define the absolute value of the relative change in weight $\delta = |\frac{\Delta W}{W}|$ to characterize the degree of change of each element in the update weight ΔW . To some extent, δ can be used to indicate the importance of each element in ΔW when inserting the new editing facts. On the one hand, a portion of the elements in the update weight \bar{W} are assumed to constitute the core components of the new editing facts. On the other hand, the remaining elements are assumed minor contributions to editing. Specifically, the top- $k\%$ elements in ΔW that change the most according to δ are considered as the principal editing information and keep their original values. While for the remaining elements in ΔW , they are treated as minor contributions to editing and set to zero for regularization. Mathematically, we have the *regularized* edit update weight $\bar{\Delta W}$ as:

$$\bar{\Delta W}_{ij} = \begin{cases} \Delta W_{ij} & \text{if } \delta_{ij} \text{ in the top-}k\%, \\ 0 & \text{else.} \end{cases} \quad (3)$$

Finally, the regularized edit update weight $\bar{\Delta W}$ is added to the original weight to derive the regularized updated weight. Essentially, RECT functions

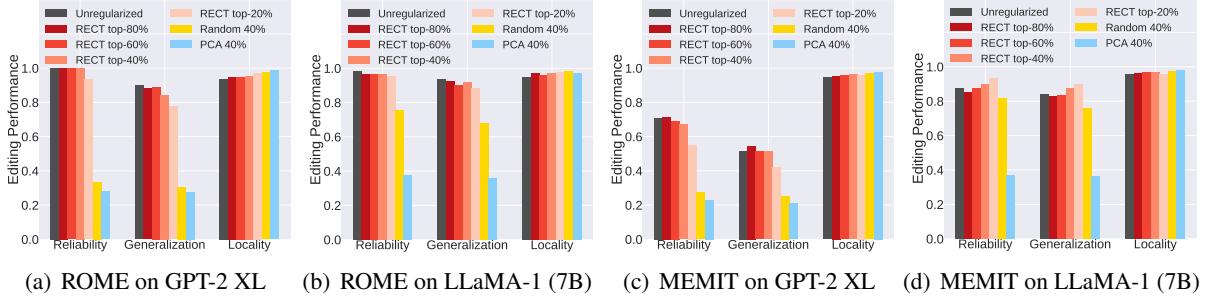


Figure 7: Comparison of introducing various regularization methods and how the *editing performance* change with respect to different top- $k\%$ for RECT.

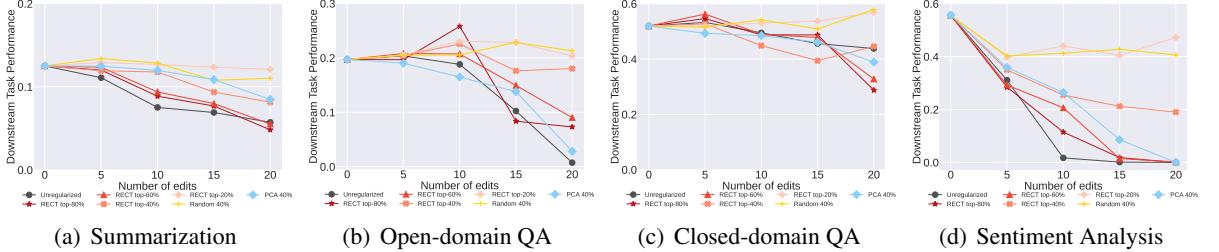


Figure 8: Comparison of introducing various regularization methods and how the *downstream task performance* change with respect to different top- $k\%$ for RECT.

to deter the implementation of excessively intricate editing updates that have a higher propensity to result in overfitting. By imposing constraints on the complexity of editing updates, it serves as a safeguard against the model’s inclination to adapt too closely to the editing data, thus promoting more generalizable and reliable model performance.

5.2 Regularization Baselines

To demonstrate the effectiveness and efficiency of the proposed method RECT, we compared it with the following baselines, including: **Unregularized** keeps the full elements of the edit update weight ΔW . **Random $k\%$** selects the random $k\%$ elements of ΔW . **PCA $k\%$** compresses the most important editing information in ΔW into $k\%$ elements via principal component analysis (PCA), and sets the remaining elements to zero.

5.3 Results of RECT

The effectiveness of a regularization method should be illustrated from two perspectives. First, regularizing the edit update weight should not harm its editing performance, i.e., edited models should still remember the new editing facts and generalize to related facts. Second, the regularized edited models should be able to preserve the general abilities compared with unregularized ones.

Editing Performance Figure 7 presents the the results of regularizing ROME or MEMIT on GPT-2 XL or LLaMA-1 (7B). Readers can refer to Appendix D.2 for the results on LLaMA-2 (7B). From these results we can have the following findings. First, compared with unregularized ΔW , RECT that keeps the original values of an appropriate amount of top-40% elements in ΔW and sets the remaining elements to zero can help maintain over 94% majority of reliability and generalization, and even improve locality. It is natural that reliability and generalization slightly drop when setting partial elements to zero since partial editing information is removed. The reason why locality is improved is probably because those elements with low δ corresponding to some noise and outliers in the editing data are removed to prevent from editing overfitting, so the edited models are more robust. However, setting excessive elements in ΔW to zero, e.g., RECT top-20%, might hurt the editing performance as partial important editing information is accidentally removed. Furthermore, compared with Random 40% and PCA 40%, RECT top-40% achieves the best performance, indicating its effectiveness in selecting the most principal editing information. It is notable that RECT also exhibits advantages in terms of efficiency, since it eliminates the complex calculations required in PCA.

General Downstream Task Performance Figure 8 presents how the downstream task performance change with respect to introducing various regularization methods to edit GPT-2 XL. Readers can refer to Appendix D.2 for the results on more downstream tasks. From these results we can have the following findings. As the proportion of elements in ΔW set to 0 increases, the more editing overfitting is regularized, the smaller the change to the original weight, so the general abilities can be more preserved. Results show that regularized edited models are able to preserve the general abilities compared with unregularized ones in most tasks such as summarization, open- and closed-domain QA. It is worth noting that it still poses a challenge for some tasks such as sentiment analysis, and remains unclear whether it works for larger number of edits, which will be left to future work.

6 Conclusion

Model updating technology has been catalyzing the continuous iteration of advanced and trustworthy LLMs. This paper studies model editing and for the first time raises concerns whether model editing has any side effects on the general abilities of LLMs. The systematical evaluation reveals that current methods unintentionally hurt the general abilities of LLMs no matter in instance- or batch-editing, and single- or sequential-editing. Our analysis of the causes reveals that model editing results in overfitting to the editing facts, and the accumulation of overfitting across multiple edits can amplify the negative impact. The proposed RECT regularization method has been proven to effectively prevent overfitting of new editing facts, thus preserving both the editing and general downstream task performance.

Impact Statements

As LLMs play an increasingly crucial role in various applications, mitigating the hallucinations caused by missing, false or outdated knowledge encapsulated within the parameters is imperative for ensuring the reliability of their outputs. However, the potential trade-off between improving the factuality and degrading the general abilities underscores the need for a balanced approach. Striking the right balance in model editing is crucial to prevent unintended consequences and to preserve the broader abilities of LLMs, contributing to the sustainable advancement of AI technology. This

paper highlights the importance of considering not only the immediate gains in factuality but also the long-term impacts on the general performance and applicability of LLMs, encouraging a thoughtful and comprehensive exploration of model editing techniques for responsible AI development. More importantly, this paper calls for more efforts and underscores the collective focus on *strengthening the robustness of LLMs to weight perturbations, developing innovative paradigms for model editing, and designing comprehensive evaluation of model editing*. By doing so, we can collectively advance the continual development of LLMs, paving the way for more reliable applications in real-world scenarios.

Limitations

This paper studies the side effects of editing based on the ZsRE editing dataset, while more complex and diverse side effects are hypothesized to exist and thus need to be explored on more editing datasets in future work. In addition, although sometimes a method of side effect mitigation is effective for a certain number of edits, it remains to be seen whether the method will still be effective for a larger number of edits. It is expected that one editing method outperforms another in terms of the number of edits given the same requirements of maintaining editing performance and general abilities. This paper does not further explore whether the proposed method can still be effective for more edits, which is worth further study. While we primarily propose to mitigate the side effects of model editing from a statistical perspective, the bottleneck of the general abilities of edited models should be analyzed theoretically.

Acknowledgement

This research is based upon work supported by an Amazon AGI foundation research award, a google research scholar grant, CISCO sponsored research award, and NSF #2331966. We thank Tanmay Parekh, Po-Nien Kung, Sidi Lu, Fabrice Harel-Canada, UCLA NLP group members and anonymous reviewers for their valuable feedback.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual*

Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. [Editing factual knowledge in language models](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6491–6506. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. [Reading wikipedia to answer open-domain questions](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1870–1879. Association for Computational Linguistics.

Siyuan Cheng, Bozhong Tian, Qingbin Liu, Xi Chen, Yongheng Wang, Huajun Chen, and Ningyu Zhang. 2023. [Can we edit multimodal large language models?](#) In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13877–13888. Association for Computational Linguistics.

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](#). *J. Mach. Learn. Res.*, 24:240:1–240:113.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. [Boolq: Exploring the surprising difficulty of natural yes/no questions](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *CoRR*, abs/2110.14168.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. [Evaluating the ripple effects of knowledge editing in language models](#). *Trans. Assoc. Comput. Linguistics*, 12:283–298.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. [Mutual: A dataset for multi-turn dialogue reasoning](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1406–1416. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. [The PASCAL recognising textual entailment challenge](#). In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer.

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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8493–8502. Association for Computational Linguistics.

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. 2023. [Erasing concepts from diffusion models](#). In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 2426–2436. IEEE.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. [SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization](#). In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. 2024. [Model editing at scale leads to gradual and catastrophic forgetting](#). In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 15202–15232. Association for Computational Linguistics.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. [Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. [Transformer-patcher: One mistake worth one neuron](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of hallucination in natural language generation](#). *ACM Comput. Surv.*, 55(12):248:1–248:38.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris

Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: a benchmark for question answering research](#). *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. [Latent retrieval for weakly supervised open domain question answering](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28–August 2, 2019, Volume 1: Long Papers*, pages 6086–6096. Association for Computational Linguistics.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. [Zero-shot relation extraction via reading comprehension](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 333–342. Association for Computational Linguistics.

Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Mengru Wang, Xi Chen, and Huajun Chen. 2024. [Unveiling the pitfalls of knowledge editing for large language models](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. [The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems](#). In *Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic*, pages 285–294. The Association for Computer Linguistics.

Jun-Yu Ma, Jia-Chen Gu, Zhen-Hua Ling, Quan Liu, and Cong Liu. 2023. [Untying the reversal curse via bidirectional language model editing](#). *CoRR*, abs/2310.10322.

Shengyu Mao, Ningyu Zhang, Xiaohan Wang, Mengru Wang, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2023. [Editing personality for llms](#). *CoRR*, abs/2310.02168.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. [Locating and editing factual associations in GPT](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022a. [Fast model editing at scale](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831. PMLR.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. [Check your facts and try again: Improving large language models with external knowledge and automated feedback](#). *CoRR*, abs/2302.12813.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. [Is chatgpt a general-purpose natural language processing task solver?](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1339–1384. Association for Computational Linguistics.

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

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. [Introduction to the conll-2003 shared task: Language-independent named entity recognition](#). In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147. ACL.

Anton Sinitcisin, Vsevolod Plokhotnyuk, Dmitry V. Pyrkin, Sergei Popov, and Artem Babenko. 2020. [Editable neural networks](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. [Recursive deep models for semantic compositionality over a sentiment treebank](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. [Llama: Open and efficient foundation language models](#). *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Cantón-Ferrer, Moya Chen, Guillem Cucurull, et al. 2023b. [Llama 2: Open foundation and fine-tuned chat models](#). *CoRR*, abs/2307.09288.

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, and Huajun Chen. 2024a. [EasyEdit: An easy-to-use knowledge editing framework for large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 82–93. Association for Computational Linguistics.

Yiwei Wang, Muhan Chen, Nanyun Peng, and Kai-Wei Chang. 2024b. [Deepedit: Knowledge editing as decoding with constraints](#). *CoRR*, abs/2401.10471.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. [Finetuned language models are zero-shot learners](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Suhang Wu, Minlong Peng, Yue Chen, Jinsong Su, and Mingming Sun. 2023a. [Eva-kellm: A new benchmark for evaluating knowledge editing of llms](#). *CoRR*, abs/2308.09954.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. 2023b. [DEPN: detecting and editing privacy neurons in pretrained language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2875–2886. Association for Computational Linguistics.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. [Editing large language models: Problems, methods, and opportunities](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 10222–10240. Association for Computational Linguistics.

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. 2024. [A comprehensive study of knowledge editing for large language models](#). *CoRR*, abs/2401.01286.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. [Siren’s song in the AI ocean: A survey on hallucination in large language models](#). *CoRR*, abs/2309.01219.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. 2023. [Mquake: Assessing knowledge editing in language models via multi-hop questions](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 15686–15702. Association for Computational Linguistics.

A Details of Evaluation Metrics

Reliability Given an editing fact ($s = \text{United States}$, $r = \text{President of}$, $o = \text{Donald Trump}$, $o^* = \text{Joe Biden}$), it could be regarded edited effectively if the edited model f^* assigns a higher probability to the statement “*The President of the United States is Joe Biden*” than the original prediction (*Donald Trump*).

Generalization Edited models should be able to recall the updated knowledge when prompted within the editing scope. For example, the paraphrased prompts like “*Who currently holds the office of President of the United States?*” or “*Who is the current president of the US?*” can be used. The edited model f^* is considered to have generalized successfully if it can recall the editing memory, in this case, “*Joe Biden*”.

Locality The edited model f^* should remain unchanged in response to prompts that are irrelevant or outside the scope of its editing. For example, the answer to the question “*Who is the President of France?*” should still correctly be “*Emmanuel Macron*”.

B Task Prompts

The prompts for each task were illustrated in Table 2.

Reasoning:

Q: {QUESTION} A: Let’s think step by step. {HINT} Therefore, the answer (arabic numerals) is:

NLI:

{SENTENCE1} entails the {SENTENCE2}. True or False? answer:

Open-domain QA:

Refer to the passage below and answer the following question. Passage: {DOCUMENT} Question: {QUESTION}

Closed-domain QA:

Please answer the given question based on the passage. The answer should be exact ‘yes’ or ‘no’. passage: {PASSAGE} question: {QUESTION}. answer:

Dialogue:

Q: {ARTICLE} Which choice is correct? Answer Choices: (A){OPTION0} (B){OPTION1} (C){OPTION2} (D){OPTION3} A: Among A through D, the answer is

Summarization:

{DIALOGUE} TL;DR:

NER:

Please identify Person Entity from the given text. Text: {SENTENCE} Entity:

Please identify Location Entity from the given text. Text: {SENTENCE} Entity:

Please identify Organization Entity from the given text. Text: {SENTENCE} Entity:

Please identify Miscellaneous Entity from the given text. Text: {SENTENCE} Entity:

Sentiment analysis:

For each snippet of text, label the sentiment of the text as positive or negative. The answer should be exact ‘positive’ or ‘negative’. text: {TEXT} answer:

Table 2: The prompts to LLMs for evaluating their zero-shot performance on these general tasks.

C Editing Methods & Datasets

Paradigm	Editing Methods	Additional Training	Batch Edit	Editor Parameters
Meta-learn	MEND	Yes	Yes	$Model_{hyper} + L*MLP$
Locate-then-edit	KN ROME MEMIT	No No No	No No Yes	$L*neuron$ MLP_{proj} $L*MLP_{proj}$

Table 3: Comparisons between several popular model editing methods following Yao et al. (2023). **Additional Training** refers to whether the methods need training before conducting specific edits. **Batch Edit** refers to editing multiple target knowledge simultaneously. **Editor Parameters** refers to the parameters that need to be updated for editing. L denotes the number of layers to update. MLP is FFN and MLP_{proj} is the second linear layer in FFN. $neuron$ denotes the key-value pair in FFN.

Four popular editing methods as compared in Table 3 were selected to measure their performance in improving factuality as well as their impairment in the general abilities of LLMs, including:

KN (Dai et al., 2022)² first selected neurons that were associated with knowledge expression via gradient-based attributions, and then modified MLP layer at the rows corresponding to those neurons by adding scaled embedding vectors.

MEND (Mitchell et al., 2022a)³ learned a hypernetwork to produce weight updates by decomposing the fine-tuning gradients into rank-1 form.

ROME (Meng et al., 2022)⁴ first localized the factual knowledge at a specific layer in the transformer MLP modules, and then updated the knowledge by directly writing new key-value pairs in the MLP module.

MEMIT (Meng et al., 2023)⁵ expanded the capabilities of ROME by enabling the editing of large amounts of factual data through the updating of a sequence of MLP layers.

All experiments were conducted using the EasyEdit tool (Wang et al., 2024a), ensuring standardized and reproducible evaluation. All editing instances were randomly sampled from the editing dataset.

The popular model editing dataset Zero-Shot Relation Extraction (ZsRE) (Levy et al., 2017) used in previous work (Cao et al., 2021; Meng et al., 2022; Yao et al., 2023) was adopted in our experiments. ZsRE is a QA dataset using question rephrasings generated by back-translation as the equivalence neighborhood. Each input is a question about an entity, and plausible alternative edit labels are sampled from the top-ranked predictions of a BART-base model trained on ZsRE.

²<https://github.com/EleutherAI/knowledge-neurons>

³<https://github.com/eric-mitchell/mend>

⁴<https://github.com/kmeng01/rome>

⁵<https://github.com/kmeng01/memit>

D Extensive Evaluation Results

D.1 Results of Side Effects of Model Editing

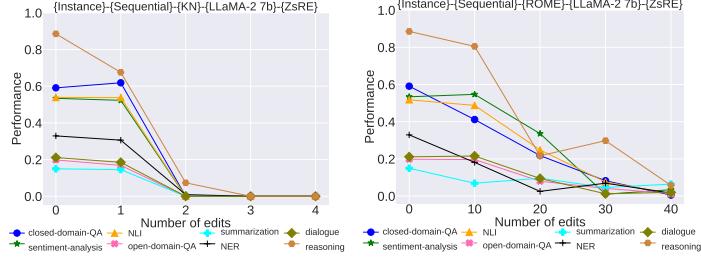


Figure 9: Performance on general tasks of edited models using KN or ROME to edit LLaMA-2 (7B) as the number of edits increases in *instance- and sequential-editing*.

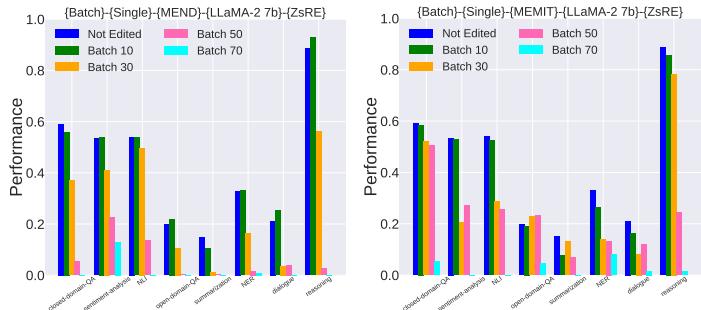


Figure 10: Performance on general tasks of edited models using MEND or MEMIT to edit LLaMA-2 (7B) with different batch sizes in *batch- and single-editing*.

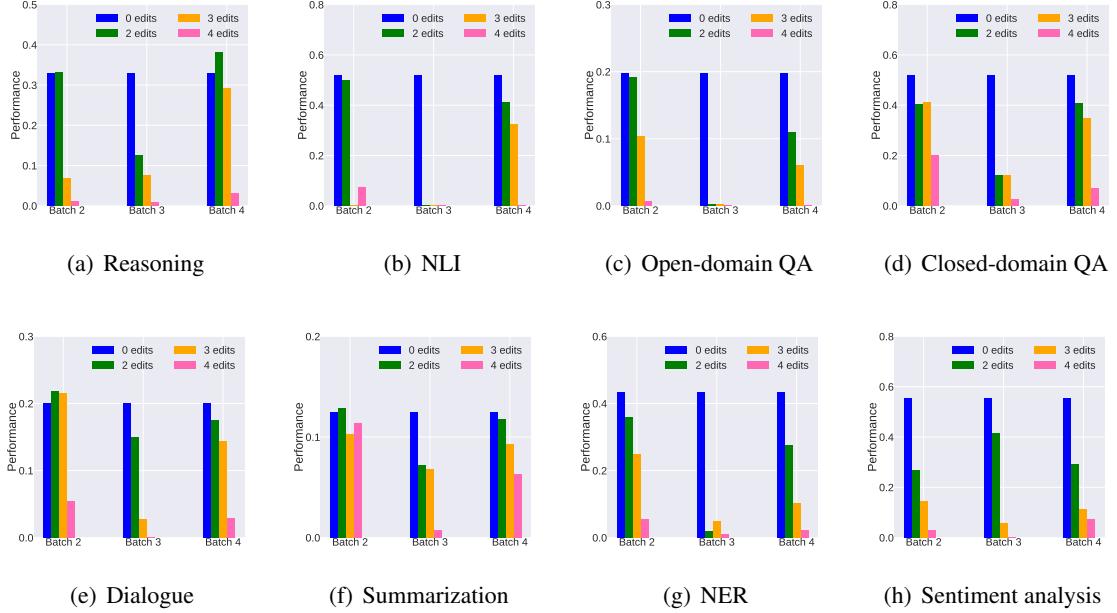


Figure 11: Performance on general tasks of edited models using MEND to edit GPT2-XL as the number of edits increases in *batch- and sequential-editing*.

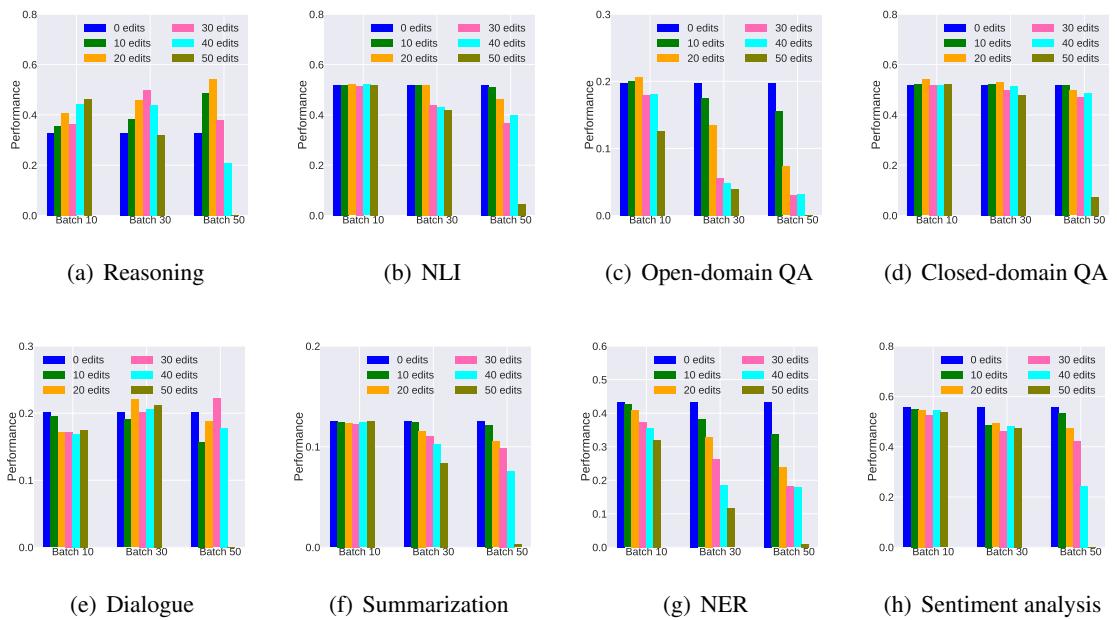


Figure 12: Performance on general tasks of edited models using MEMIT to edit GPT2-XL as the number of edits increases in *batch- and sequential-editing*.

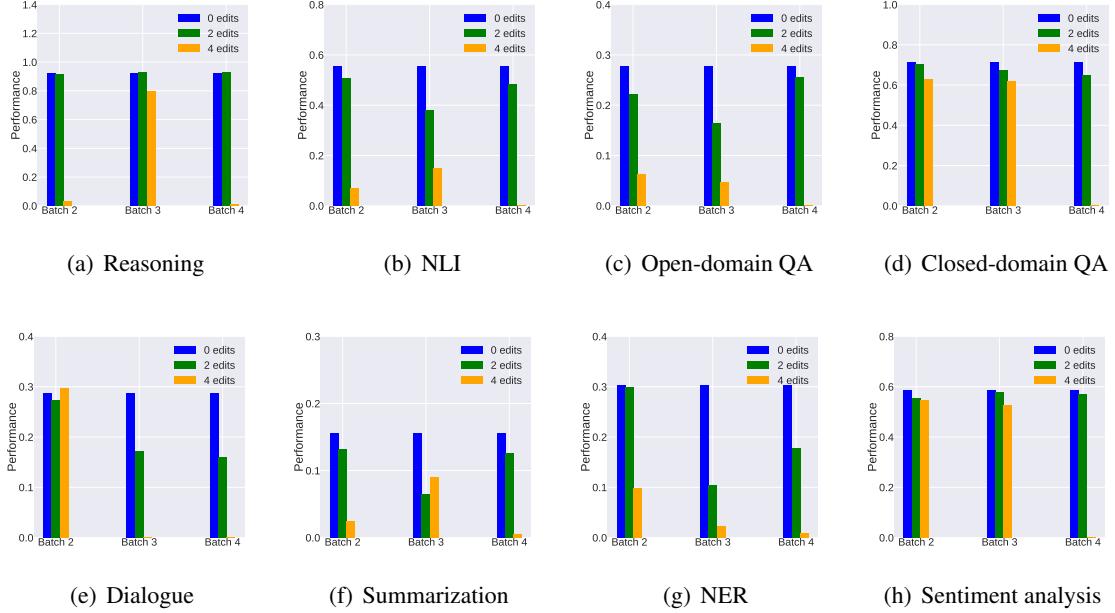


Figure 13: Performance on general tasks of edited models using MEND to edit LLaMA-1 (7B) as the number of edits increases in *batch- and sequential-editing*.

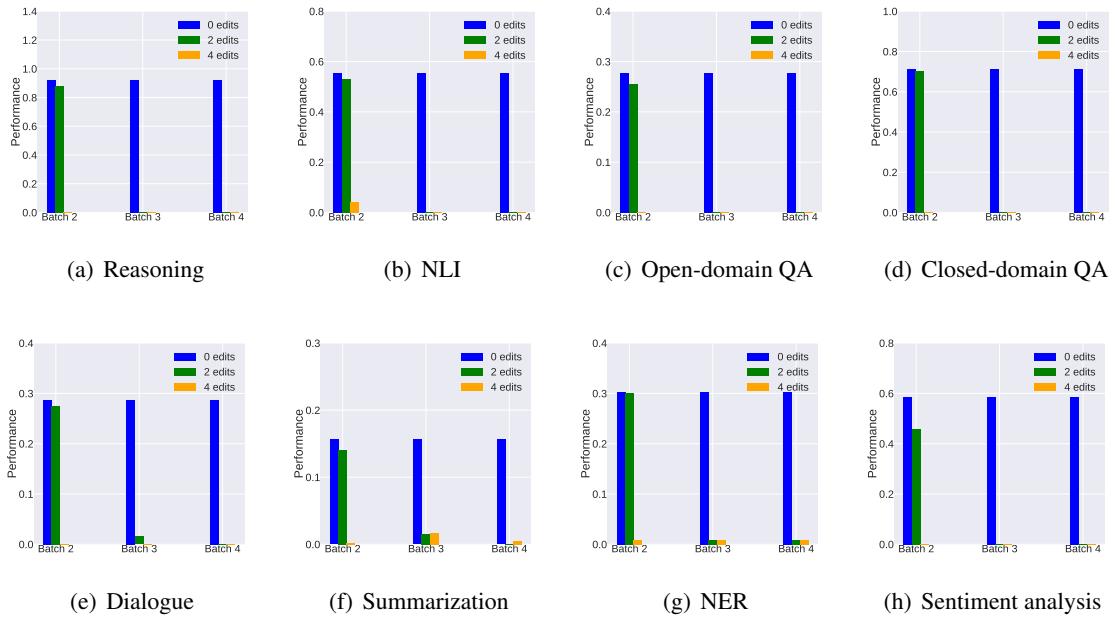
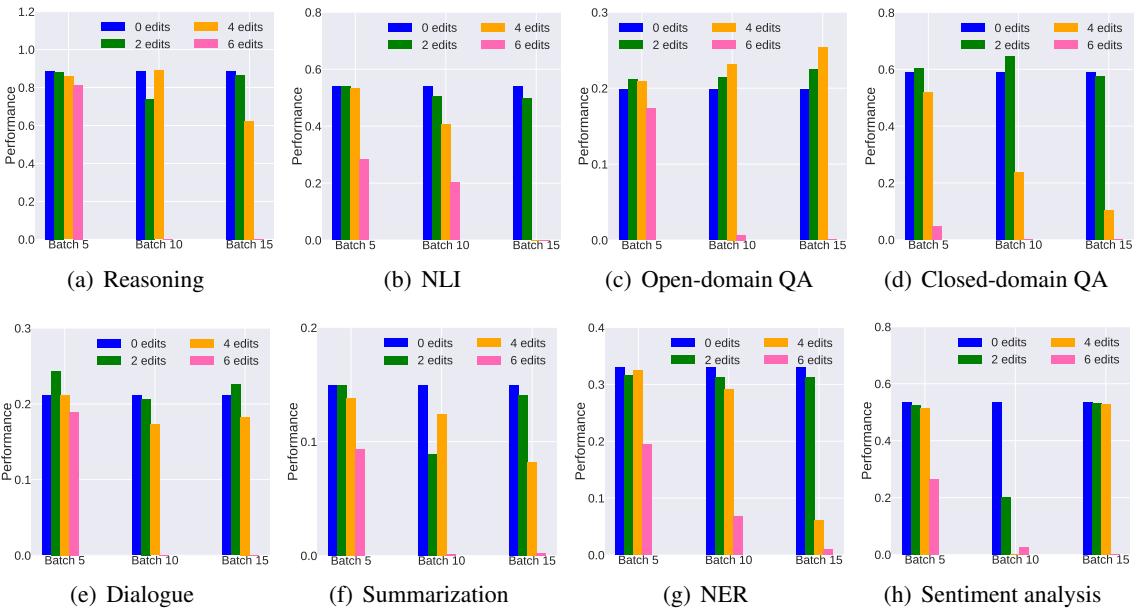
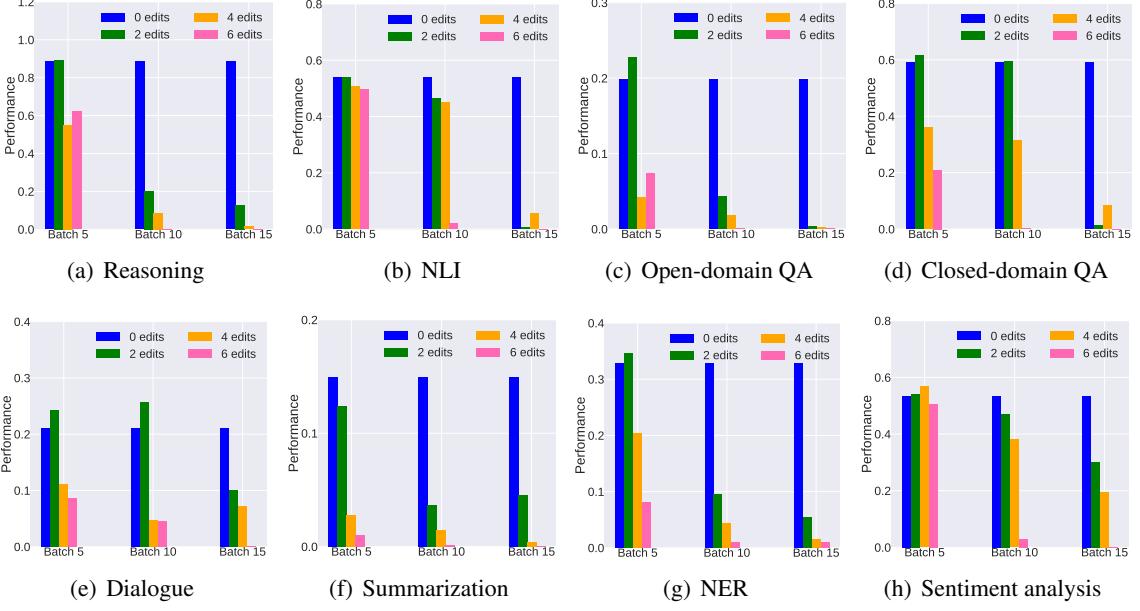
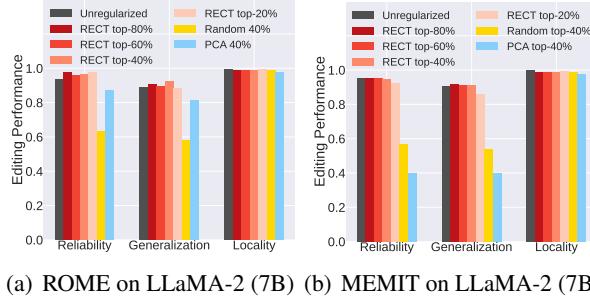


Figure 14: Performance on general tasks of edited models using MEMIT to edit LLaMA-1 (7B) as the number of edits increases in *batch- and sequential-editing*.



D.2 Results of RECT



(a) ROME on LLaMA-2 (7B) (b) MEMIT on LLaMA-2 (7B)

Figure 17: Comparison of introducing various regularization methods and how the *editing performance* change with respect to different top- $k\%$ for RECT.

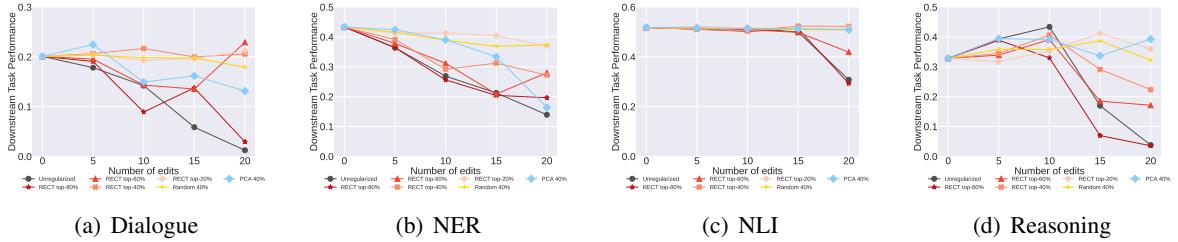


Figure 18: Comparison of introducing various regularization methods for ROME and how the *downstream task performance* change with respect to different top- $k\%$ for RECT based on GPT-2 XL.