# Reasons and Solutions for the Decline in Model Performance after Editing

**Xiusheng Huang**[*] [1,2,3], **Jiaxiang Liu**[*] [1,2], **Yequan Wang**[†] [3]
**Kang Liu**[†] [1,2]
[1]The Key Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]Beijing Academy of Artificial Intelligence, Beijing, China
huangxiusheng2020@ia.ac.cn, liujiaxiang21@mails.ucas.ac.cn,
tshwangyequan@gmail.com, kliu@nlpr.ia.ac.cn

## Abstract

Knowledge editing technology has received widespread attention for low-cost updates of incorrect or outdated knowledge in large-scale language models. However, recent research has found that edited models often exhibit varying degrees of performance degradation. The reasons behind this phenomenon and potential solutions have not yet been provided. In order to investigate the reasons for the performance decline of the edited model and optimize the editing method, this work explores the underlying reasons from both data and model perspectives. Specifically, 1) from a data perspective, to clarify the impact of data on the performance of editing models, this paper first constructs a **M**ulti-**Q**uestion **D**ataset (**MQD**) to evaluate the impact of different types of editing data on model performance. The performance of the editing model is mainly affected by the diversity of editing targets and sequence length, as determined through experiments. 2) From a model perspective, this article explores the factors that affect the performance of editing models. The results indicate a strong correlation between the L1-norm of the editing model layer and the editing accuracy, and clarify that this is an important factor leading to the bottleneck of editing performance. Finally, in order to improve the performance of the editing model, this paper further proposes a **D**ump **for S**equence (**D4S**) method, which successfully overcomes the previous editing bottleneck by reducing the L1-norm of the editing layer, allowing users to perform multiple effective edits and minimizing model damage. Our code is available at https://github.com/nlpkeg/D4S.

## 1 Introduction

Large-scale language models (LLMs) have demonstrated exceptional performance in NLP tasks [Huang et al., 2021, 2022]. However, as knowledge continues to evolve, LLMs inevitably contain incorrect or outdated information. Due to their vast number of parameters, directly fine-tuning the model would require a substantial amount of computational resources [Gupta et al., 2023]. As a result, knowledge editing techniques have emerged as a low-cost and effective method for updating a model's knowledge [Yao et al., 2023]. These techniques involve modifying a small number of the model's parameters to update its internal knowledge [Ding et al., 2023]. However, growing evidence suggests that altering model parameters can have a negative impact on the model's performance. The

[*]Equal contribution.
[†]Corresponding authors.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-2199

specific reasons behind this phenomenon remain unclear, and corresponding optimization methods are currently unavailable.

Previous research has investigated the performance degradation of edited models [Wang et al., 2023, Mazzia et al., 2023]. Specifically, Hase et al. [2024] found that edited models suffer from catastrophic forgetting, where they forget previously edited samples. Additionally, Gu et al. [2024] showed that edited models exhibit significant performance declines on downstream tasks, which severely hinders the practical applicability of knowledge editing techniques. However, these studies failed to identify the underlying causes of performance degradation in edited models and did not propose optimization methods to mitigate this issue [Zhang et al., 2024].

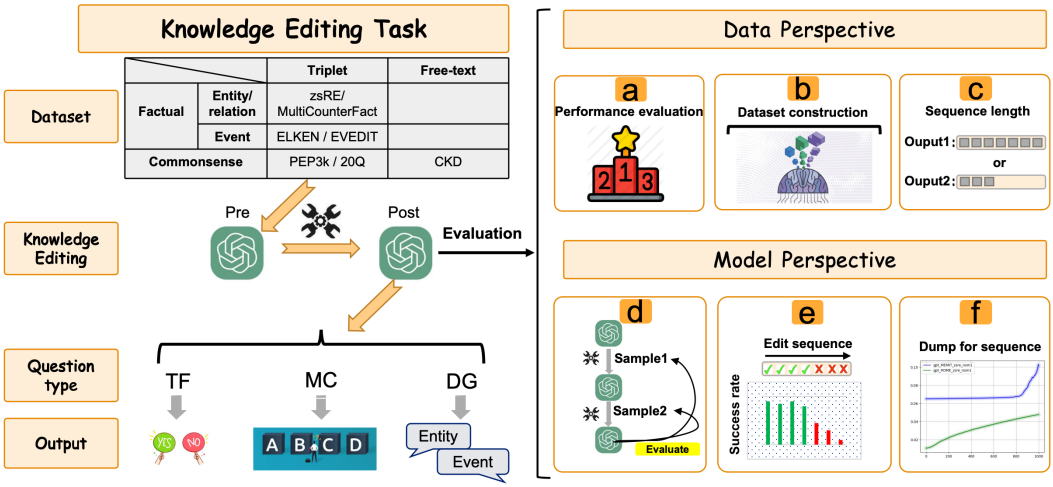



Fig. 1: This framework outlines the comprehensive approach to understanding the performance decline of edited models. On the left, traditional knowledge editing tasks are categorized into different types, each with distinct editing objectives: yes/no, a/b/c/d, and entity/event. On the right, our experiments are structured from both data and model perspectives. From the data perspective, we conduct three experiments: (a) a comprehensive performance evaluation of the model, (b) the construction of a **M**ulti-**Q**uestion **D**ataset (MQD), and (c) an assessment of the impact of editing different target outputs on model performance. From the model perspective, we design four experiments: (d) an evaluation of the edited model's forgetting ability, (e) an identification of the current knowledge editing method's bottleneck and an exploration of the correlation between editing probability values and parameter layer norms, and (f) a proposal of a sequence editing method, which effectively enhances the performance of the edited model.

This paper examines the factors that influence model performance and optimization methods from both data and model perspectives. As shown in Figure 1, from the data perspective, we evaluated the performance of edited models on multiple generalization tasks and found that the performance degradation of edited models is correlated with the editing objectives. We then constructed **M**ulti-**Q**uestion **D**ataset (MQD) with different question types, including multiple-choice, true/false, and direct generation, with corresponding editing objectives of yes/no, a/b/c/d, and entity/event, respectively. By calculating the perplexity (PPL) of different editing objectives, we discovered that the larger the PPL, the more severe the performance degradation of the edited model.

From the model's perspective, we investigate the reasons behind the decline in model performance from two angles: catastrophic forgetting and the bottleneck imposed by the number of edits. Our analysis reveals a strong correlation between the accuracy of edits and the L1-norm growth of the parameter layers after editing. To mitigate this issue, we propose a **D**ump **for** **S**equence (D4S) method that regulates the explosive growth of parameter layers during the editing process. This approach effectively enhances the performance of the edited model and overcomes the bottleneck associated with the number of edits.

To the best of our knowledge, we are the first to investigate the causes of performance degradation in edited models and concurrently propose an effective sequence editing method to enhance the performance of edited models. Our contributions can be summarized as follows:

- To investigate the impact of data on the performance of edited models, we performed evaluations across multiple tasks, revealing that the editing objective is the primary factor influencing model performance. By creating datasets with diverse question types, we established a strong correlation between the perplexity (PPL) of the editing objectives and the performance of the edited model.
- We find that the decline in edited model performance is correlated with the explosive growth of the L1-norm of parameter layers during the editing process, which is the primary cause of both catastrophic forgetting and the bottleneck imposed by the number of edits.
- To enhance the performance of the edited model, we propose a caching sequence edit method that leverages $\mathcal{O}(1)$ space complexity to retain past knowledge, regulates the explosive growth of parameter layer norms, and thereby effectively improves the performance of the edited model, ultimately overcoming the bottleneck imposed by the number of edits.

## 2 Related Work

### 2.1 Knowledge Editing Datasets

The existing knowledge editing dataset can be divided into triplet form and event form. In triplet format dataset, commonsense knowledge dataset includes PEP3k and 20Q [Porada et al., 2021, Gupta et al., 2023], factual knowledge includes ZsRE [Levy et al., 2017], CounterFact [Meng et al., 2022a], Fact Verification [Mitchell et al., 2022] , Calibration [Dong et al., 2022], MQuAKE [Zhong et al., 2023] and RaKE [Wei et al., 2023]. In event format dataset, datasets with only factual knowledge, including ELKEN [Peng et al., 2024], EVEDIT [Liu et al., 2024] and CKD[Huang et al., 2024].

### 2.2 Knowledge Editing Methods

The previous editing methods mainly focused on editing knowledge in the form of triples, with a small amount of knowledge in the form of editing events. The methods for editing triplet forms mainly include : (1)Locate-Then-Edit method [Dai et al., 2021, Meng et al., 2022a,b, Li et al., 2024], (2) Memory-based method [Mitchell et al., 2022, Madaan et al., 2022, Zhong et al., 2023, Zheng et al., 2023], (3) Hyper-network method [Mitchell et al., 2021, De Cao et al., 2021, Tan et al., 2023]. The method for editing event forms is Self-Edit [Liu et al., 2024].

### 2.3 Model Evaluation after Editing

The damage caused to the model by updating model parameters is unknown. [Hase et al., 2024] found that the edited model had catastrophic forgetting issues and performance degradation on downstream tasks. [Gu et al., 2024] evaluated the edited model on eight downstream tasks and used multiple editing methods. It was found that the edited model exhibited varying degrees of performance degradation. However, the above methods only found a decrease in performance of the edited model, without pointing out the reasons for the performance decline. At the same time, they did not propose effective editing methods to improve the performance of the edited model.

## 3 Data-Specific Factors Affecting Performance

In this section, we investigate the primary data factors contributing to the decline in model performance following editing.

### 3.1 The Overall Performance Evaluation

To assess the performance of the edited model, we curated a diverse range of editing and evaluation datasets for testing.

**The Dataset for Editing.** As illustrated in Figure 1, we selected common sense knowledge and factual knowledge datasets as editing corpora, featuring diverse data formats such as triplets and free text. In Figure 2, we utilized the zsRE[Levy et al., 2017], ELKEN[Peng et al., 2024], 20Q[Porada et al., 2021, Gupta et al., 2023], and CKD[Huang et al., 2024] datasets as editing corpora. Notably,

 https://doi.org/10.52202/079017-2199

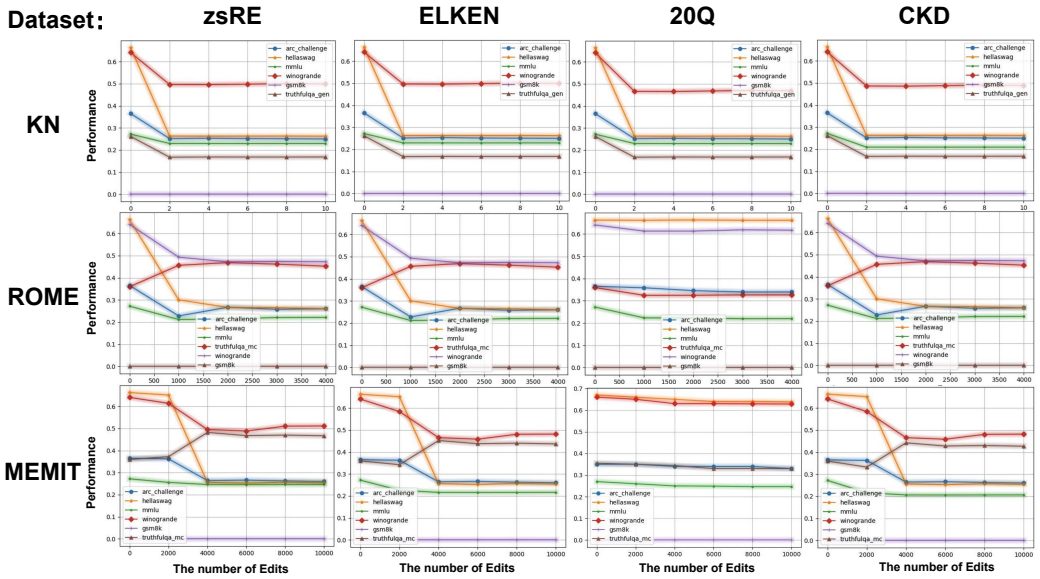



Fig. 2: Evaluation results of different editing methods on various types of datasets. The horizontal axis in the image represents the number of edited samples, and the vertical axis represents the performance of the edited model.

the commonsense dataset 20Q has editing objectives in the form of 0/1 labels, whereas the other three datasets have editing objectives of entity, entity, and event, respectively.

**The Dataset for Evaluation.** We employed six types of datasets in the Open LLM Leadership board evaluation, comprising AI2 Reasoning Challenge (25-shot) [Clark et al., 2018], HellaSwag (10-shot) [Zellers et al., 2019], the Common Sense Reasoning Dataset, MMLU (5-shot) [Hendrycks et al., 2020], a widely used benchmark for assessing multitask accuracy, which encompasses 57 tasks including basic mathematics, American history, computer science, law, and others. Additionally, we utilized TruthfulQA (0-shot) [Lin et al., 2021], a benchmark designed to test the model's propensity for dishonesty, as well as WinoGrande [Sakaguchi et al., 2021]: Common Sense Reasoning and GSM-8K [Cobbe et al., 2021]: Mathematical Reasoning.

**Result Analysis.** As illustrated in Figure 2, we observed that the KN [Dai et al., 2021] editing method substantially degrades the model's performance after editing a single sample. In contrast, when employing the ROME [Meng et al., 2022a] editing method, the model's performance did not experience a significant decline even after editing 1000 samples. Similarly, with the MEMIT [Meng et al., 2022b] method, the model's performance remained relatively stable when editing up to 2000 samples. However, beyond 2000 samples, a pronounced decline in performance occurred, and further increases in the number of edited samples did not lead to additional performance degradation.

It is noteworthy that the ROME [Meng et al., 2022a] and MEMIT [Meng et al., 2022b] methods exhibit less pronounced performance degradation when editing the 20Q [Porada et al., 2021, Gupta et al., 2023] dataset. In contrast, when editing other datasets, the model's performance displays varying degrees of decline. This can be attributed to the fact that the 20Q [Porada et al., 2021, Gupta et al., 2023] dataset's editing objective is in the form of binary labels (0/1), featuring a single element for the editing objective and a relatively small number of tokens.

### 3.2 Dataset Construction

To elucidate the impact of different editing objectives on the performance of the edited model, we created a **M**ulti-**Q**uestion **D**ataset (MQD) based on the ATOMIC commonsense database [Sap et al., 2019]. This dataset comprises three question types: true/false, multiple-choice, and direct generation. The corresponding editing objectives are yes/no, a/b/c/d, and entity/event, respectively. Each question type consists of 4000 samples.

| ATOMIC Data Source: < PersonX accepts PersonY appointment, as a result, PersonY shakes PersonX hand > | | |
|---|---|---|
| **Category** | **Component Prompt** | **Target Answer** |
| DG | PersonX accepts PersonY appointment, resulting in PersonY | shakes PersonX hand |
| MQ | PersonX accepts PersonY appointment, resulting in PersonY ?<br>Below are four options: (a) to give him a treat; (b) to forgive him;<br>(c) to live happily ever after; (d) shakes PersonX hand. The correct option is | d |
| T/F | PersonX accepts PersonY appointment, resulting in PersonY<br>shakes PersonX hand . Is this sentence logical? Please answer yes or no. A: | yes |

Table 1: An example of converting source data from ATOMIC [Sap et al., 2019] database into directly generated(DG), multiple-choice questions(MQ), and true/false questions(T/F). The editing objectives are "shakes PersonX hand", "d" and "yes", respectively.

**Data Preparation.** The ATOMIC database [Sap et al., 2019], developed by the Allen Institute and later optimized in its subsequent version [Hwang et al., 2021], is a well-known commonsense repository. The data format in ATOMIC [Sap et al., 2019] is $< \mathrm{Event}_1, \mathrm{Relationship}, \mathrm{Event}_2 >$, which contains unrecognized markers (e.g. ___, etc.) and invalid characters (e.g., &, etc.) that we manually filtered out. Furthermore, the relationship types in ATOMIC [Sap et al., 2019] are abbreviated and not easily comprehensible by humans. Although ATOMIC [Sap et al., 2019] provides corresponding annotations, they are insufficient to form a coherent statement when constructing the prompt. To address this, we also performed manual template rewriting to ensure a smoother overall prompt.

**MQD Dataset.** Different problem formats are associated with distinct editing objectives. The MQD dataset encompasses three formats: Directly Generated (DG), Multiple-choice Questions (MQ), and True/False questions (T/F). According to our statistical analysis, the Perplexity (PPL) values for the editing objectives of these three question types are 12.3, 43.3, and 297.4, respectively. The calculation formula for PPL is as follows:

$$\mathrm{PPL}(X) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta\left(x_i \mid x_{<i}\right)\right\} \tag{1}$$

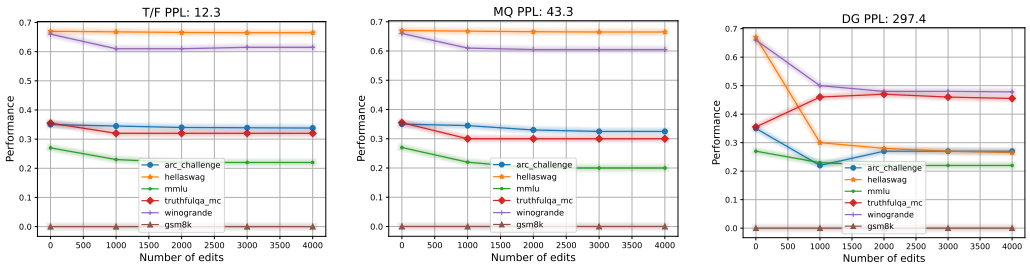
Where $p_\theta\left(x_i \mid x_{<i}\right)$ is based on the sequence before i, and the log-likelihood of the i-th token. In the Table 1, for directly generating datasets, we directly concatenate $\mathrm{Event}_1$, the rewritten relationship, and $\mathrm{Event}_2$ to form a prompt. For multiple-choice questions, we designated the target answer as one of the options and randomly selected events from other natural categories in the dataset as the remaining options. For true/false questions, we asked LLMs to determine whether the newly formed prompt is logical. To mitigate bias, we established multiple positive and negative examples. Notably, the core information of the three question types remains consistent, with the primary difference lying in the question types and editing objectives.

### 3.3 The Influence of Editing Objectives

In this section, we investigate the effect of various editing objectives on model performance by editing the **M**ulti-**Q**uestion **D**ataset (MQD)3.2.

**The Dataset for Editing and Evaluation.** The MQD dataset comprises three question types: true/false, multiple-choice, and directly generated. The knowledge sources for these three question types are consistent, and we conducted relevant experiments directly using the MQD dataset. We selected ROME [Meng et al., 2022a] as the editing method. According to our statistical analysis, the average length of the input tokens for the three question types is 23.44, 35.03, and 13.38, respectively, while the average length of the editing objectives tokens is 1, 1, and 3.88, respectively. The true/false questions have two possible output types: yes or no. The multiple-choice questions have four editing objectives: a, b, c, and d. In contrast, the directly generated questions have more diverse editing objectives, including entities or events, with the number of tokens for events typically exceeding 1.


https://doi.org/10.52202/079017-2199




(a) True/False questions (T/F). (b) Multiple-choice (MQ). (c) Directly generated (DG).

Fig. 3: The performance of the model after editing data for different question types.

**Result Analysis.** As illustrated in Figure 3, all three question types - true/false, multiple-choice, and direct generation - compromise the model's performance, with direct generation causing the most significant damage. The experimental results suggest that a higher perplexity (PPL) of the editing objectives leads to more severe performance degradation. Meanwhile, we also calculated the L1-norm of the editing layer and found that the higher the perplexity of the editing target, the larger the L1-norm. Consequently, **from a data perspective, the decline in model performance after editing is attributed to the diversity of editing objectives and the length of tokens.**

## 4 Model-Specific Factors Affecting Performance

In this section, we investigate the model-centric factors contributing to performance degradation and propose optimization strategies to enhance the performance of the edited model.

### 4.1 Forgetting about Previously Edited Samples

As illustrated in Figure 7 in Appendix B, conventional editing methods typically assess a sample's quality immediately after editing. However, in real-world applications, it is often necessary to evaluate the editing success rate after processing a sequence of samples. The traditional evaluation approach overlooked the edited model's impact on previously edited samples. This section investigates the model's forgetting issue with respect to previously edited samples.

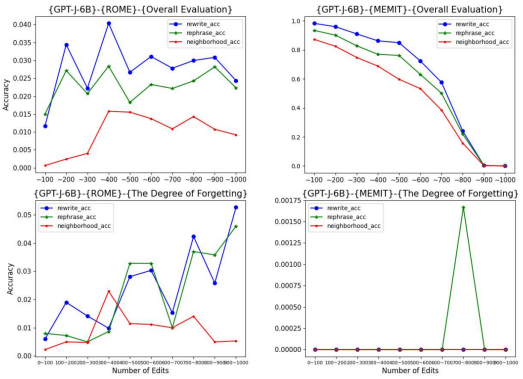



Fig. 4: Assessment of forgetting ability of models.

**The dataset and method for evaluation.** We employed the factual triplet dataset zeRE[Levy et al., 2017] for experimental purposes and utilized ROME [Meng et al., 2022a] and MEMIT [Meng et al., 2022b] as editing methods. To assess the varying degrees of forgetting in the model, we devised two experimental approaches. As illustrated in Figure 4, the "Overall Evaluation" involves evaluating all previous samples using the edited model each time it is completed. In contrast, "The Degree of Forgetting" involves obtaining a checkpoint after editing 1000 samples, followed by an evaluation of the model's forgetting degree every 100 previously edited samples, in the order of editing.

**Result analysis.** As shown in Figure 4, for the "Overall Evaluation", the ROME [Meng et al., 2022a] method has the highest accuracy of 0.04, indicating that the edited model has a significant forgetting problem on the 100 previously edited samples. The accuracy of the MEMIT [Meng et al., 2022b] method has decreased from the highest near 1.0 to 0.0, indicating that as the number of model edits increases, the forgetting problem of the model becomes increasingly severe. For the degree of forgetting, the ROME [Meng et al., 2022a] method and MEMIT [Meng et al., 2022b] method always have a low success rate. Especially the MEMIT [Meng et al., 2022b] method shows a success rate of 0 in most samples. We suspect that the MEMIT [Meng et al., 2022b] method after editing 1000

https://doi.org/10.52202/079017-2199

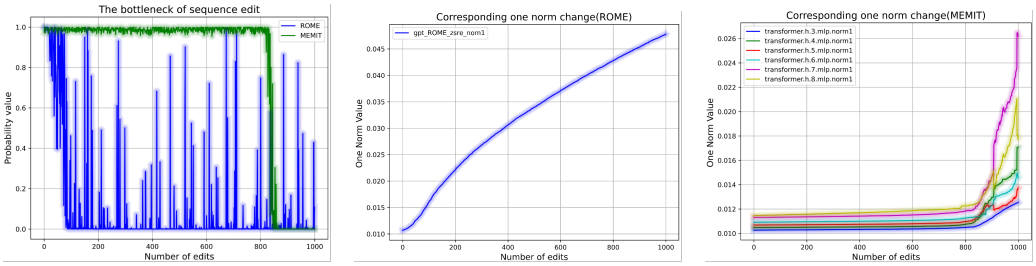
samples may not be able to successfully edit the samples. We also conducted relevant experiments in the next section.

### 4.2 The Bottleneck of Sequence Edit

Sequence editing refers to the repeated updating of knowledge within a model. However, the number of successful edits that existing knowledge editing methods can achieve is not infinite. We conducted experiments on two editing methods to examine the bottleneck in the number of edits for each method.

**The dataset and method for evaluation.** The editing models used in the experiments are all based on GPT-J (6B) [Wang and Komatsuzaki, 2021]. Consistent with the original work, the fixed MLP layers for the ROME [Meng et al., 2022a] and PMET [Li et al., 2024] methods are [5] and [3,4,5,6,7,8], respectively. We used the factual triplet dataset zeRE for experimental data and applied ROME and MEMIT as editing methods. All experiments were conducted with a batch size of 1, and a total of 1000 samples were edited. We recorded the probability value of each edited sample.

**Result analysis.** As shown in Figure 5.a, after editing 100 samples, the ROME method shows an overall decrease in probability values. While some probability values still update to larger values as the number of edits increases, the overall trend is downward. For the MEMIT method, after editing 850 samples, the probability values showed a significant decrease, approaching zero. This indicates that the MEMIT method exhibits a phenomenon of complete editing inefficiency after 850 edits. This experiment reflects the bottleneck phenomenon in editing methods, with ROME having an effective editing count of 100 and MEMIT having an effective editing count of 850.




(a) The probability value. (b) L1-norm on ROME. (c) L1-norm on MEMIT.

Fig. 5: The bottleneck and L1-norm correspondence in sequence editing are illustrated in the figure. In subfigure (a), the horizontal axis represents the number of edited samples, and the vertical axis represents the probability value of the edited object. The blue line represents the ROME method, while the green line represents the MEMIT method. In subfigures (b) and (c), the horizontal axis represents the number of edits, and the vertical axis represents the L1-norm value of the editing layer.

We plotted the L1-norm variation of different methods on the editing layer during the editing process. As shown in Figure 5.b, when the number of edits reaches 100, the norm of the 5-th MLP layer edited by the ROME method significantly increases. Furthermore, in Figure 5.c, after 850 edits using the MEMIT method, the editing layers [3, 4, 5, 6, 7, 8] also exhibit an explosive increase in norm. The results indicate that **from a model perspective, the decline in model performance after editing is due to the explosive growth of norms in the editing layers during the editing process.**

### 4.3 Dump for Sequence Knowledge Editing

Sequence editing refers to the process of editing a single or multiple samples multiple times. With the continuous updating of world knowledge, constantly updating the knowledge within models has become an urgent need. The experimental results in Sections 3 and 4 show that the performance of the model significantly decreases after sequence editing. In this section, we propose a **D**ump **for** **S**equence (D4S) knowledge editing method that effectively improves the performance of the edited model.


https://doi.org/10.52202/079017-2199

#### 4.3.1 Defects in Previous Sequence Editing

The knowledge editing method is to update factual associations stored in the parameters of transformers[Meng et al., 2022b]. Given a sequence of text $x = [x_1, \cdots, x_m]$, the transformer's hidden state $h^{l,j}$ at the layer $l$ and the token $j$ is calculated:

$$\begin{aligned} h^{l,j}[x] &= h^{l-1,j}[x] + att^{l,j}[x] + m^{l,j}[x] \\ att^{l,j}[x] &= attention^{l}(h^{l-1,1}[x], \cdots, h^{l-1,j}[x]) \\ m^{l,j}[x] &= W^{l}_{out}\sigma(W^{l}_{in}\gamma(att^{l,j}[x])) \end{aligned} \tag{2}$$

where $\gamma$ indicate layer norm and $\sigma$ means a non-linear function. The knowledge editing method is to update the knowledge in the model by changing particular weight $W$. For MEMIT [Meng et al., 2022b] method, there are triples of fact to be edited $\xi = \{(s_i, r_i, o_i)\}_{i=1}^{n}$, where $s_i$ means the subject, $o_i$ is the object and $r_i$ means relation between them. And we have $o_i \neq LLM(s_i, r_i)$. For simplicity, we use $h^{l}_{i}[x]$ to represent the last token's hidden state ($h^{l,m}_{i}[x]$) of the $i^{th}$ prompt $x$. To make $o_i = LLM^{'}(s_i, r_i)$, the target of $i^{th}$ edit $z_i = h^{L}_{i} + \delta_i$ is got by optimizing:

$$\min_{\delta_i} \frac{1}{N}\sum_{k=1}^{N} -logP[o_i|pre_k \oplus p(s_i, r_i)] \tag{3}$$

where $h^{L}_{i} = h^{L}_{i}[p(s_i)]$ is the hidden state of the last edit layer $L$, $p(s_i, r_i)$ denotes the prompt consisting of subject and relation, and $pre_k$ indicates a random prefix to obtain generalizability. Subsequently, for each layer $l \in \mathcal{L}$ needs to be edited, we can take the following approach to update $W^{l}_{out} \in \mathcal{R}^{u \times v}$:

$$k^{l}_{i} = \frac{1}{N}\sum_{k=1}^{N} \sigma(W^{l}_{in}\gamma(h^{l-1}_{i}[pre_k \oplus p(s_i)])), r^{l}_{i} = \frac{z_i - h^{L}_{i}}{L - l + 1} \tag{4}$$

$$\Delta^{l} = (R^{l}{K^{l}}^{T})(Cov^{l} + K^{l}{K^{l}}^{T})^{-1}, W^{l}_{out} \leftarrow W^{l}_{out} + \Delta^{l} \tag{5}$$

with $R^{l} = [r^{l}_{1}, \cdots, r^{l}_{n}]$, $K^{l} = [k^{l}_{1}, \cdots, k^{l}_{n}]$, $Cov^{l} = K^{l}_{0}{K^{l}_{0}}^{T}$. And $K^{l}_{0}$ are the keys of knowledge irrelevant with $\xi$. A simple idea to optimize the knowledge of sequence editing as a whole is saving the editing history $R^{l}$ and $K^{l}$. For each new edit with $k^{l}_{n+1}$ and $r^{l}_{n+1}$, we can concate it with history:

$${R^{l}}^{'} = R^{l} \oplus r^{l}_{n+1} = [r^{l}_{1}, \cdots, r^{l}_{n}, r^{l}_{n+1}] \tag{6}$$

$${K^{l}}^{'} = K^{l} \oplus k^{l}_{n+1} = [k^{l}_{1}, \cdots, k^{l}_{n}, k^{l}_{n+1}] \tag{7}$$

In this way, we can optimize all the knowledge of sequence editing as a whole to mitigate the damage to LLM. However, the space complexity of such a dump method is $\mathcal{O}(n)$, which is unacceptable to us.

#### 4.3.2 The D4S Method

The D4S method is designed to save the editing history in $\mathcal{O}(1)$ space and apply batch editing methods in sequence editing situations. We note that for $\Delta^{l} = (R^{l}{K^{l}}^{T})(Cov^{l} + K^{l}{K^{l}}^{T})^{-1}$, the parts related to the editing history can be written as:

$$R^{l}{K^{l}}^{T} = [r^{l}_{1}, \cdots, r^{l}_{n}][k^{l}_{1}, \cdots, k^{l}_{n}]^{T} = \sum_{i=1}^{n} r^{l}_{i}{k^{l}_{i}}^{T} \tag{8}$$

$$K^{l}{K^{l}}^{T} = [k^{l}_{1}, \cdots, k^{l}_{n}][k^{l}_{1}, \cdots, k^{l}_{n}]^{T} = \sum_{i=1}^{n} k^{l}_{i}{k^{l}_{i}}^{T} \tag{9}$$

where we have ${R^l}{K^l}^T \in \mathcal{R}^{u\times v}$ and ${K^l}{K^l}^T \in \mathcal{R}^{v\times v}$. So we just need to save the two matrices above. For each new edit with $k^l_{n+1}$ and $r^l_{n+1}$, we can integrate it into edit history with a smiple addition operation:

$$ {R^l}{K^l}^{T'} = {R^l}{K^l}^T + r^l_{n+1}{k^l_{n+1}}^T \tag{10} $$

$$ {K^l}{K^l}^{T'} = {K^l}{K^l}^T + k^l_{n+1}{k^l_{n+1}}^T \tag{11} $$

This approach requires just $\mathcal{O}(1)$ storage space and allows us to convert sequence editing methods into batch editing methods, thus reducing the damage to the edited model during sequence editing. Additionally, this ability to consolidate each individual edit instance into a single batch makes the locate and editing method distinct from fine-tuning.

#### 4.3.3 Theoretical Proof of Mitigating Norm Growth

To demonstrate that our D4S method can effectively alleviate norm growth in the editing layer, we can consider the update of parameters edited by the previous method MEMIT [Meng et al., 2022b] after $n$ edits:

$$ \Delta W_{MEMIT} = \sum_{i=1}^{n}(r_i k_i^T)(K_0 K_0^T + k_i k_i^T)^{-1} \tag{12} $$

Regarding the D4S method, we have:

$$ \Delta W_{D4S} = (\sum_{i=1}^{n} r_i k_i^T)(K_0 K_0^T + \sum_{i=1}^{n} k_i k_i^T)^{-1} = \sum_{i=1}^{n}(r_i k_i^T)(K_0 K_0^T + \sum_{i=1}^{n} k_i k_i^T)^{-1} \tag{13} $$

Due to $K_0 K_0^T + k_i k_i^T$ and $K_0 K_0^T + \sum_{i=1}^{n} k_i k_i^T$ being positive definite, intuitively, the inverse of $K_0 K_0^T + \sum_{i=1}^{n} k_i k_i^T$ is expected to have smaller numerical values compared to $K_0 K_0^T + k_i k_i^T$. Therefore, the norm of $\Delta W_{D4S}$ is smaller than that of $\Delta W_{MEMIT}$. The experimental results in Figures 6 also demonstrate the effectiveness of the D4S method in mitigating L1-norm growth.

## 5 Experiments

The experiments are designed to answer two research questions: a) How does D4S perform in sequence editing compared to other methods? b) How much damage does D4S do to the model?

### 5.1 Performance Comparison of Sequence Editing

We counted the results of different methods after 1,000 sequence edits on GPT-J (6B) [Wang and Komatsuzaki, 2021] and Llama2 (7B) [Touvron et al., 2023]. The Appendix C provides Metric indicators. As show in Table 2, compared to previous editing methods, our method D4S has achieved significant performance improvement. Specifically, when Edits=500 and Edits=1000, the average performance of D4S achieved State-Of-The-Art(SOTA) results, indicating that D4S still has superior editing ability even after editing multiple samples.

We conduct additional experiments on Counterfact[Meng et al., 2022b] and Mquake[Zhong et al., 2023] dataset in Appendix D. In addition to this, we also wanted to explore the model's forgetting of previous edits, so we chose every 100 edits as a checkpoint to investigate this in Appendix E.

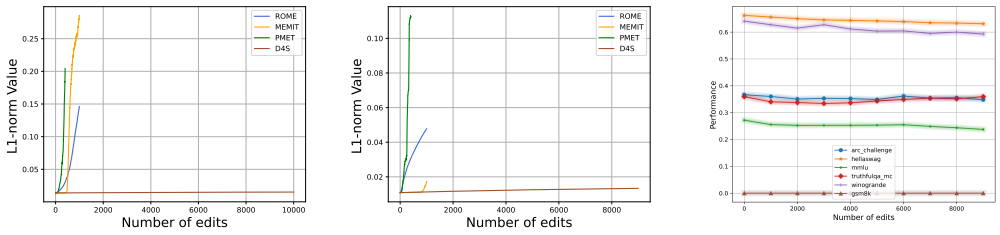

https://doi.org/10.52202/079017-2199

| Model | Method | $Edits = 100$ | | | | $Edits = 500$ | | | | $Edits = 1000$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $Eff.$ | $Par.$ | $Spe.$ | $Avg.$ | $Eff.$ | $Par.$ | $Spe.$ | $Avg.$ | $Eff.$ | $Par.$ | $Spe.$ | $Avg.$ |
| Llama | FT | 30.47 | 25.78 | 12.19 | 22.81 | 19.43 | 18.06 | 6.56 | 14.68 | 16.13 | 13.89 | 5.96 | 12.00 |
| | ROME | 97.68 | **92.39** | 90.22 | **93.43** | 48.68 | 47.72 | 32.56 | 42.99 | 21.84 | 20.74 | 4.01 | 15.53 |
| | MEMIT | 92.99 | 83.22 | 94.02 | 90.08 | 2.90 | 2.90 | 2.51 | 2.77 | 2.85 | 2.85 | 2.74 | 2.82 |
| | PMET | 0.24 | 0.35 | 0.46 | 0.35 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | GRACE | **99.33** | 0.53 | **99.88** | 66.58 | **98.87** | 0.59 | **99.72** | 66.39 | **98.94** | 0.35 | **99.77** | 66.35 |
| | **D4S**(ours) | 95.53 | 86.45 | 93.44 | 91.81 | 91.19 | **81.48** | 83.34 | **85.34** | 87.36 | **79.30** | 78.99 | **81.88** |
| GPT | FT | 15.28 | 7.40 | 0.42 | 7.70 | 12.95 | 7.84 | 0.23 | 7.01 | 7.03 | 4.15 | 0.11 | 3.77 |
| | ROME | 1.17 | 1.50 | 0.06 | 0.91 | 2.67 | 1.83 | 1.55 | 2.02 | 2.44 | 2.23 | 0.92 | 1.86 |
| | MEMIT | 98.25 | 93.41 | 87.24 | 92.97 | 84.94 | 76.15 | 59.72 | 73.60 | 0.00 | 0.02 | 0.00 | 0.01 |
| | PMET | 0.33 | 0.33 | 0.00 | 0.22 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | GRACE | **100.00** | 0.40 | **100.00** | 66.8 | **100.00** | 0.15 | **100.00** | 66.72 | **100.00** | 0.07 | **100.00** | 66.69 |
| | **D4S**(ours) | 97.72 | **97.01** | 85.30 | **93.34** | 97.42 | **93.12** | 74.65 | **88.40** | 95.98 | **91.17** | 70.17 | **88.77** |

Table 2: Sequencial edits on GPT and Llama with *ZsRE* dataset. The best results are in **bold** and underline means the suboptimal.

### 5.2 Performance of Edited Model

As shown in Figure 6, we explore the impact of D4S on the model by examining the changes in average weight norms and performance on downstream tasks of GPT. The downstream task performance changes for Lllama are in the Appendix F. The experimental results indicate that the norm of the D4S method did not increase after 10000 edits, and the performance of downstream tasks only decreased slightly.




(a) Llama_L1-Norm (b) GPT_L1-Norm (c) Downstream tasks

Fig. 6: Norms of weight and performance of the edited model.

## 6 Conclusion

This paper explores the reasons behind the decline in model performance from both data and model perspectives. From the data perspective, the decline is attributed to the diversity of editing objectives and the length of tokens. From the model perspective, the decline is due to the explosive growth of norms in the editing layers during the editing process. To enhance the performance of edited models, we propose a **D**ump **for** **S**equence (D4S) method, which effectively improves the performance of edited models and overcomes previous editing bottleneck issues. This method allows for multiple effective edits with minimal impact on model performance.

## Acknowledgments

This work was supported by Beijing Natural Science Foundation (L243006) and the National Science Foundation of China (No. 62106249). This work was also sponsored by CCF-BaiChuan-Ebtech Foundation Model Fund.

## References

Xiusheng Huang, Yubo Chen, Shun Wu, Jun Zhao, Yuantao Xie, and Weijian Sun. Named entity recognition via noise aware training mechanism with data filter. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4791–4803, 2021.

Xiusheng Huang, Hang Yang, Yubo Chen, Jun Zhao, Kang Liu, Weijian Sun, and Zuyu Zhao. Document-level relation extraction via pair-aware and entity-enhanced representation learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2418–2428, 2022.

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

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

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey. *ACM Computing Surveys*, 2023.

Vittorio Mazzia, Alessandro Pedrani, Andrea Caciolai, Kay Rottmann, and Davide Bernardi. A survey on knowledge editing of neural networks. *arXiv preprint arXiv:2310.19704*, 2023.

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

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. Model editing can hurt general abilities of large language models. 2024. URL `https://doi.org/10.48550/arXiv.2401.04700`.

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

Ian Porada, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. Modeling event plausibility with consistent conceptual abstraction. *arXiv preprint arXiv:2104.10247*, 2021.

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

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

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

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. Calibrating factual knowledge in pretrained language models. *arXiv preprint arXiv:2210.03329*, 2022.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*, 2023.

Yifan Wei, Xiaoyan Yu, Huanhuan Ma, Fangyu Lei, Yixuan Weng, Ran Song, and Kang Liu. Assessing knowledge editing in language models via relation perspective. *arXiv preprint arXiv:2311.09053*, 2023.

Hao Peng, Xiaozhi Wang, Chunyang Li, Kaisheng Zeng, Jiangshan Duo, Yixin Cao, Lei Hou, and Juanzi Li. Event-level knowledge editing. *arXiv preprint arXiv:2402.13093*, 2024.

Jiateng Liu, Pengfei Yu, Yuji Zhang, Sha Li, Zixuan Zhang, and Heng Ji. Evedit: Event-based knowledge editing with deductive editing boundaries. *arXiv preprint arXiv:2402.11324*, 2024.

 https://doi.org/10.52202/079017-2199

Xiusheng Huang, Yequan Wang, Jun Zhao, and Kang Liu. Commonsense knowledge editing based on free-text in llms. 2024.

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

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

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. Pmet: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18564–18572, 2024.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. Memory-assisted prompt editing to improve gpt-3 after deployment. *arXiv preprint arXiv:2201.06009*, 2022.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning? *arXiv preprint arXiv:2305.12740*, 2023.

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

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

Chenmien Tan, Ge Zhang, and Jie Fu. Massive editing for large language models via meta learning. *arXiv preprint arXiv:2311.04661*, 2023.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035, 2019.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. On symbolic and neural commonsense knowledge graphs. 2021.

Ben Wang and Aran Komatsuzaki. Gpt-j-6b: A 6 billion parameter autoregressive language model, 2021.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

## A Appendix /Limitations

We note a few limitations of the experiments conducted in this paper:

(1) This paper only used the GPT-J (6B) and Llama2 (7B) models. Due to limitations in computational resources, experiments on larger scale models were not conducted.

(2) For the D4S method, our training consists of only 2000 steps due to computational resource limitations. By editing more samples, we can better identify the performance bottleneck of the D4S method.

(3) Due to computational resource limitations, the MEMIT and PMET methods did not use a batch size greater than 1 in our experiments. To better assess batch editing, more experiments with larger batch sizes should be attempted.

## B Appendix /Evaluation Method

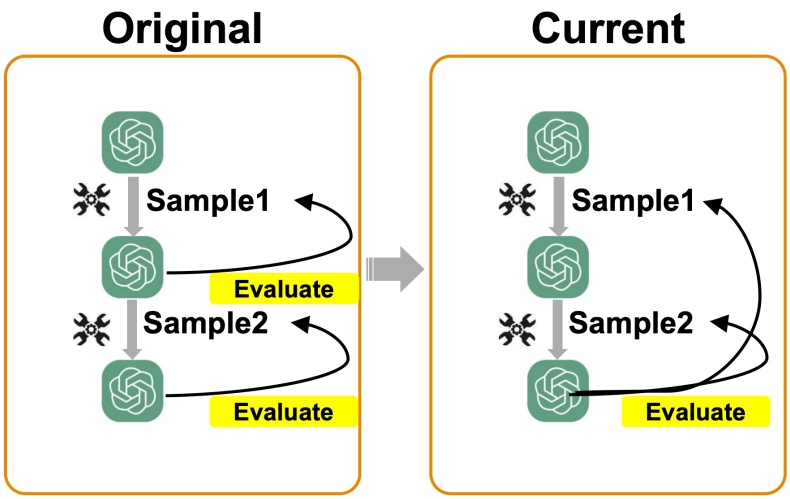



Fig. 7: The original and current evaluation methods.

## C Appendix /Metric Indicators

Same as MEMIT, we took the following three indicators to evaluate the impact on model performance:

$$Eff. = \mathbb{E}_i[o_i = \underset{o_i^*}{argmax}\, P[o_i^*|p(s_i, r_i)]] \tag{14}$$

$$Par. = \mathbb{E}_i[\mathbb{E}_{p \in par(s_i, r_i)}[o_i = \underset{o_i^*}{argmax}\, P[o_i^*|p]] \tag{15}$$

$$Spe. = \mathbb{E}_i[\mathbb{E}_{(s_i^e, r_i^e, o_i^e) \in nei(s_i, r_i, 0_i)}[o_i^e = \underset{o_i^*}{argmax}\, P[o_i^*|p(s_i^e, r_i^e)]] \tag{16}$$

where $par(s_i, r_i)$ is the set of $p(s_i, r_i)$'s paraphrases, $nei(s_i, r_i, o_i)$ is the set of $(s_i, r_i, o_i)$'s neighborhoods.

## D Appendix /Additional Experiments

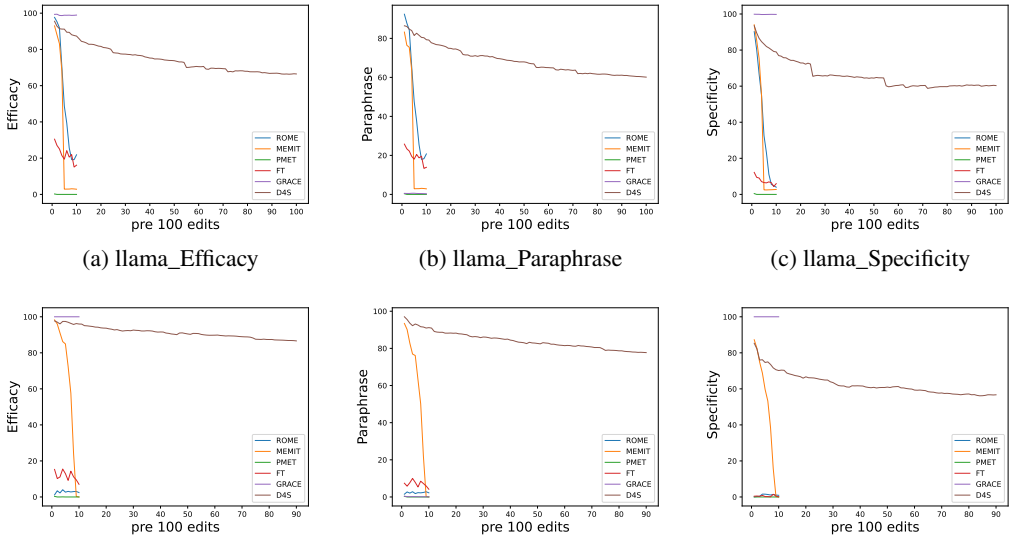
We also conduct additional experiments on Counterfact[Meng et al., 2022b] and Mquake[Zhong et al., 2023] dataset. Here are the results:


https://doi.org/10.52202/079017-2199

| Model | Method | *Counterfact* | | | | *Mquake* | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | *Eff.* | *Par.* | *Spe.* | *Avg.* | *Eff.* | *Par.* | *Spe.* | *Avg.* |
| GPT | FT | 32.80 | 9.00 | 1.00 | 14.27 | 17.00 | 6.22 | 0.00 | 7.74 |
| | ROME | 0.60 | 0.70 | 0.60 | 0.63 | 0.00 | 0.00 | 0.00 | 0.00 |
| | MEMIT | 86.20 | **59.50** | 31.30 | 59.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | GRACE | **100.00** | 0.50 | **100.00** | 66.83 | - | - | - | - |
| | **D4S**(ours) | 99.10 | 47.00 | 63.70 | **69.93** | **97.40** | **75.54** | **21.84** | **64.93** |
| Llama | FT | 8.46 | 4.07 | 2.03 | 4.85 | 41.43 | 44.93 | **28.24** | 38.20 |
| | ROME | 27.83 | 16.03 | 5.66 | 16.50 | 76.85 | **78.41** | 3.67 | 52.97 |
| | MEMIT | 0.00 | 0.00 | 6.72 | 2.24 | 0.00 | 0.00 | 0.00 | 0.00 |
| | GRACE | **99.9** | 0.25 | **99.97** | 66.71 | - | - | - | - |
| | **D4S**(ours) | 96.68 | **46.66** | 72.45 | **71.93** | **85.30** | 72.68 | 28.16 | **62.05** |

Table 3: Sequencial edits on GPT and Llama with *Counterfact* and *Mquake* dataset. The best results are in **bold** and underline means the suboptimal. Since GRACE is not suitable for the data structure of *Mquake*, we did not include it in the comparison.

## E Appendix /Evaluate Forgetting Ability




(a) llama_Efficacy (b) llama_Paraphrase (c) llama_Specificity

(d) gpt_Efficacy (e) gpt_Paraphrase (f) gpt_Specificity

Fig. 8: Evaluation metrics change with the number of edits.

## F Appendix /Downstream Performance of Llama

| Edit Num | arc_challenge | hellaswag | mmlu | truthfulqa_mc | winogrande |
|---|---|---|---|---|---|
| 0 | 43.34 | 57.14 | 41.35 | 38.97 | 69.14 |
| 10,000 | 42.41 (↓0.93) | 52.99 (↓4.15) | 39.76 (↓1.59) | 38.56 (↓0.41) | 68.75 (↓0.39) |

Table 4: Downstream performance of llama after 10,000 edits.

https://doi.org/10.52202/079017-2199

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: [NA]


https://doi.org/10.52202/079017-2199

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification:[NA]

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).


https://doi.org/10.52202/079017-2199

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: [NA]

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper poses no such risks.


https://doi.org/10.52202/079017-2199

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.


https://doi.org/10.52202/079017-2199