Multiplicative Orthogonal Sequential Editing for Language Models

Hao-Xiang Xu[1,2*], **Jun-Yu Ma**[1,2*], **Ziqi Peng**[1*], **Yuhao Sun**[1], **Zhen-Hua Ling**[1,2], **Jia-Chen Gu**[3†]

[1] University of Science and Technology of China
[2]National Engineering Research Center of Speech and Language Information Processing
[3]University of California, Los Angeles
{nh2001620,mjy1999,aisis,syh3327}@mail.ustc.edu.cn, zhling@ustc.edu.cn, gujc@ucla.edu

Abstract

Knowledge editing aims to efficiently modify the internal knowledge of large language models (LLMs) without compromising their other capabilities. The prevailing editing paradigm, which appends an update matrix to the original parameter matrix, has been shown by some studies to damage key numerical stability indicators (such as condition number and norm), thereby reducing editing performance and general abilities, especially in sequential editing scenario. Although subsequent methods have made some improvements, they remain within the additive framework and have not fundamentally addressed this limitation. To solve this problem, we analyze it from both statistical and mathematical perspectives and conclude that multiplying the original matrix by an orthogonal matrix does not change the numerical stability of the matrix. Inspired by this, different from the previous additive editing paradigm, a multiplicative editing paradigm termed **M**ultiplicative **O**rthogonal **S**equential **E**diting (MOSE) is proposed. Specifically, we first derive the matrix update in the multiplicative form, the new knowledge is then incorporated into an orthogonal matrix, which is multiplied by the original parameter matrix. In this way, the numerical stability of the edited matrix is unchanged, thereby maintaining editing performance and general abilities. We compared MOSE with several current knowledge editing methods, systematically evaluating their impact on both editing performance and the general abilities across three different LLMs. Experimental results show that MOSE effectively limits deviations in the edited parameter matrix and maintains its numerical stability. Compared to current methods, MOSE achieves a 12.08% improvement in sequential editing performance, while retaining 95.73% of general abilities across downstream tasks.

1 Introduction

While large language models (LLMs) have demonstrated impressive capabilities (Dubey, Jauhri, and et al. 2024), they frequently suffer from hallucinations caused by inaccurate or obsolete knowledge stored in their parameters (Zhang et al. 2023). Since retraining LLMs requires substantial time and resources, researchers have increasingly focused on *knowledge editing* (a.k.a., *model editing*) (Mitchell et al. 2022b; Meng et al. 2022, 2023; Wang et al. 2024a; Ma et al. 2024). Current approaches to knowledge editing can be roughly categorized into *parameter-modifying* methods (Meng et al. 2022, 2023) that directly modify a small subset of model parameters, or *parameter-preserving* methods (Wang et al. 2024b) that integrate additional modules without altering the model parameters. In this paper, we study the parameter-modifying editing methods.

Sequential knowledge editing (Yao et al. 2023) allows LLMs to continually integrate new knowledge through consecutive updates. In this scenario, existing editing methods predominantly follow an additive paradigm, directly modifying model parameters by adding update matrices. However, previous studies (Hu et al. 2024; Ma et al. 2025) have demonstrated that this additive editing paradigm leads the edited parameter matrix to deviate substantially from its original structure, severely damaging critical aspects of numerical stability, including the norm (Kahan 2013) and condition number (Sun 2000). These disruptions degrade the model's editing performance and general abilities. Although methods like RECT (Gu et al. 2024), EAC (Xu et al. 2025), PRUNE (Ma et al. 2025), and AlphaEdit (Fang et al. 2025) offer some mitigation, they only marginally extend the scope of editing and still cause damage to the numerical stability of the edited matrix, failing to fundamentally resolve the limitations inherent in the additive editing paradigm. This impacts model scalability and presents major challenges for continuous learning in LLMs.

To address this challenge, we pose the following question: *"Can we develop an editing paradigm that does not damage the numerical stability of the edited matrix, thereby maintaining both the editing performance and general abilities of the model?"* Orthogonal transformations are numerically stable linear operations defined by orthogonal matrices. They encode rich semantic information by changing the angle of the vector (Liu et al. 2018). In this paper, from a statistical perspective, compared to the previous editing paradigm based on additive updates, we first show that left-multiplying the matrix with an orthogonal matrix can strictly maintain its numerical stability indicator, including norm and condition number. Furthermore, a rigorous mathematical analysis is provided demonstrating that such orthogonal transformations theoretically keep both the norm and the condition number of the matrix unchanged.

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

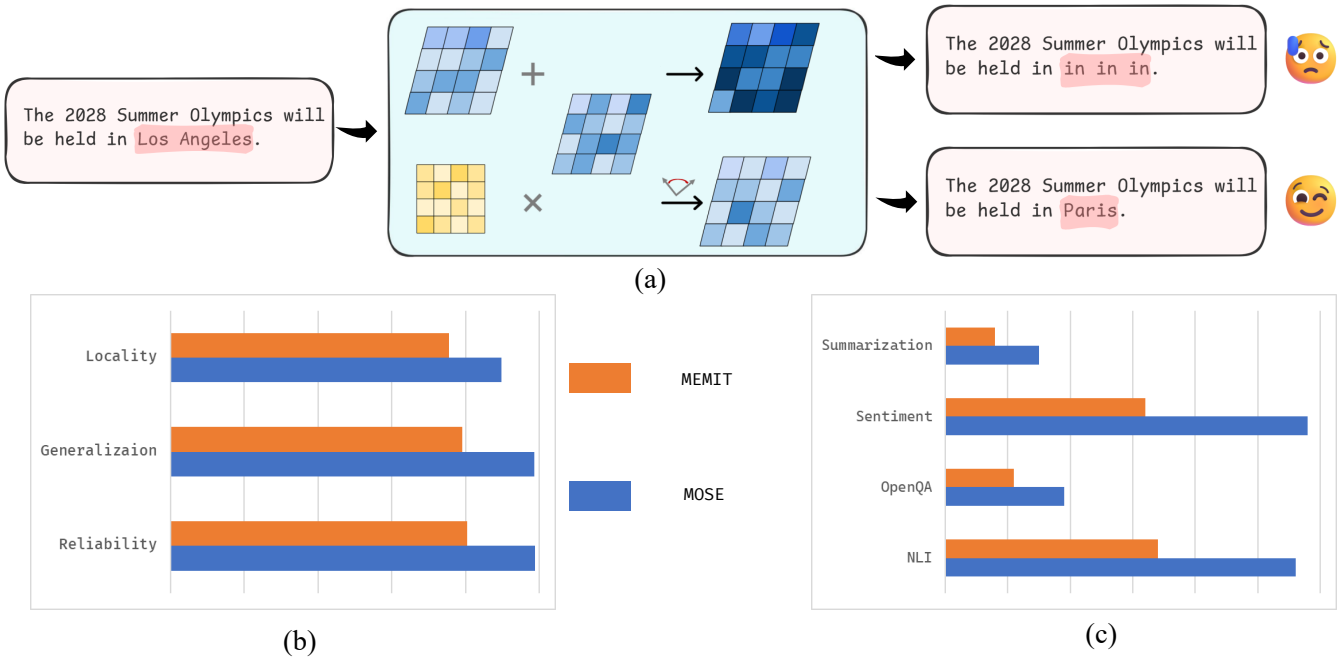

Figure 1: (a) Comparison between the previous methods and MOSE. The previous methods performed updates by adding an update matrix, whereas the MOSE employs left-multiplication by an orthogonal update matrix. (b) Comparison of editing performance after additive-based editing and after multiplicative-based editing by MOSE. (c) Comparison of general downstream task performance before editing, after additive-based editing, and after multiplicative-based editing by MOSE.

Inspired by these findings, we propose **M**ultiplicative **O**rthogonal **S**equential **E**diting (MOSE), a multiplicative editing paradigm that leverages orthogonal transformations. Specifically, MOSE introduces an orthogonal transformation update strategy, which injects new knowledge into an orthogonal update matrix and multiplies it with the original parameter matrix to perform knowledge editing, as shown in Figure 1. To achieve this, MOSE explicitly minimizes the output error for both the target knowledge to be edited and the existing knowledge to be preserved. By performing orthogonal transformation operations on the original matrix, the edited matrix not only stores the new knowledge but also does not affect its numerical stability. This ensures that the knowledge within the model is not disturbed, thereby preserving the editing performance and the general abilities.

To validate the effectiveness of MOSE, our study comprehensively evaluates the edited LLMs for both editing performance and general abilities in sequential editing. We evaluate MOSE against **six popular knowledge editing methods**, including ROME (Meng et al. 2022), MEMIT (Meng et al. 2023), RECT (Gu et al. 2024), EMMET (Gupta, Baskaran, and Anumanchipalli 2024), PRUNE (Ma et al. 2025) and AlphaEdit (Fang et al. 2025), across **three LLMs of varying sizes**, such as LLaMA3-8B (Dubey, Jauhri, and et al. 2024), LLaMA2-13B (Touvron, Martin, and et al. 2023) and Qwen2.5-7B (Yang et al. 2024a). **Three editing datasets** across two types of editing data are selected to comprehensively evaluate the impact of knowledge editing on the editing performance of LLMs and **four representative tasks** are chosen to thoroughly assess how knowledge editing affects the general abilities of these models. The experimental results demonstrate that during sequential editing, MOSE surpasses existing approaches by 12.08%, while still retaining 95.73% of the general abilities.

In summary, our contributions to this paper are three-fold: (1) This paper analyzes the key factor influencing both the editing performance and the general abilities of models after sequential editing, focusing on deviations in the parameter matrix and its numerical stability. (2) A method termed MOSE is proposed, which leverages orthogonal transformation to constrain deviations in the edited matrix and maintain its numerical stability. (3) It is found that on models of different sizes, MOSE demonstrates a 12.08% improvement in editing performance compared to existing methods, while effectively preserving over 95.73% of the model's general abilities on downstream tasks.

2 Related Work

Knowledge Editing Current knowledge editing methods typically adopt either parameter modification or preservation strategies. *Parameter-Modifying Methods* directly adjust model weights to inject new knowledge. Meta-learning approaches like KE (Cao, Aziz, and Titov 2021), MEND (Mitchell et al. 2022a), and InstructEdit (Zhang et al. 2024) use hypernetworks. Locate-then-edit methods, such as ROME (Meng et al. 2022), compute updates via normal equations, while MEMIT (Meng et al. 2023) scales this to batch editing. *Parameter-Preserving Methods* retain original weights through auxiliary designs. ICE (Zheng et al. 2023)

uses in-context learning; SERAC (Mitchell et al. 2022b) employs external memory; T-Patcher (Huang et al. 2023) and CaliNet (Dong et al. 2022) add editing-specific neurons. GRACE (Hartvigsen et al. 2023) replaces hidden states with codebooks, while WISE (Wang et al. 2024a) integrates parameterized memory to enhance editing performance.

Sequential Editing Recent research systematically investigates the challenges in sequential model editing. Gupta, Baskaran, and Anumanchipalli (2024) identifies ROME's dual key vector design as a cause of editing failure, a finding extended by Yang et al. (2024b), who attributes it to distributional discrepancies. Hu et al. (2024) shows that representation overlap in whitening space hinders editing, while Ma et al. (2025) theoretically links matrix condition numbers to editing instability. In response, recent methods propose targeted solutions: RECT (Gu et al. 2024) applies sparse updates to limit parameter drift; PRUNE (Ma et al. 2025) constrains condition numbers; AlphaEdit (Fang et al. 2025) leverages null-space constraints for near-lossless edits; and EAC (Xu et al. 2025) compresses editing anchors to preserve general abilities of the edited model.

Compared with previous studies (Gu et al. 2024; Ma et al. 2025; Xu et al. 2025; Fang et al. 2025) that are the most relevant to our work, a main difference should be highlighted. These approaches follow an additive paradigm, directly modifying model parameters by adding update matrices. However, this paradigm inevitably compromises the numerical stability of the edited matrix, ultimately degrading the model's editing performance and general abilities. In contrast, our approach systematically leverages the numerical stability of orthogonal transformations by multiplying the original parameter matrix with an orthogonal update matrix. This operation preserves the edited matrix's numerical stability while maintaining strong editing performance and general abilities across multiple sequential edits.

3 Preliminary

Knowledge editing enables targeted modification of knowledge encoded in language models (LMs) without full retraining, allowing adaptation to specific task requirements. This process facilitates precise updates to diverse learned representations, including but not limited to logical relationships, spatial understanding, and numerical reasoning. In this work, we focus on structural knowledge editing represented as (x_e, y_e)[1] pairs. Formally, given a language model $f_\theta \in \mathcal{F}$ that implements a mapping $f_\theta : \mathcal{X} \to \mathcal{Y}$ from inputs $x \in \mathcal{X}$ to predictions $y \in \mathcal{Y}$, editing aims to transform parameters $\theta \in \Theta$ to θ' such that $f_{\theta'}(x_e) = y_e$ when $f_\theta(x_e) \neq y_e$. The sequential editing extension involves an iterative process: for an edit set $\mathcal{E} = \{(x_{ei}, y_{ei}) \mid i = 1, \ldots, n\}$ and initial model f_{θ_0}, each step applies an editing function K yielding $f_{\theta_i} = K(f_{\theta_{i-1}}, (x_{ei}, y_{ei}))$.

The effects of knowledge editing generally propagate through a neighborhood of inputs that are semantically connected to the modified knowledge, which is referred to *editing scope*. To be considered successful, an editing operation must demonstrate two key properties: precise modification of outputs within this defined scope, and strict preservation of the model's behavior on all out-of-scope queries:

$$f_{\theta_i}(x_{ei}) = \begin{cases} y_{ei} & \text{if } x_{ei} \in I(x_{ei}, y_{ei}), \\ f_{\theta_{i-1}}(x_{ei}) & \text{if } x_{ei} \in O(x_{ei}, y_{ei}). \end{cases}$$

The *in-scope* $I(x_{ei}, y_{ei})$ typically includes x_{ei} and its equivalence neighborhood $N(x_{ei}, y_{ei})$, which encompasses related input/output pairs. In contrast, the *out-of-scope* $O(x_{ei}, y_{ei})$ comprises inputs unrelated to the edit example. Following established literature (Cao, Aziz, and Titov 2021; Mitchell et al. 2022a; Meng et al. 2022, 2023; Yao et al. 2023), we evaluate edits along three key dimensions: *reliability*, *generalization*, and *locality*.

[1]Can be also represented as knowledge triple $t = (subject, relation, object)$.

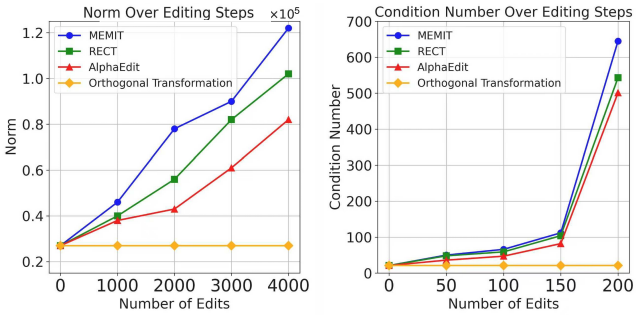

Figure 2: Illustration of the change of Frobenius norm and condition number in sequential editing at the edited layer using additive-based methods and orthogonal transformations. We selected LLaMA3-8B and CounterFact for experiments.

4 Analysis of Performance Degradation

In this section, we statistically compare the additive editing methods with the orthogonal matrix left-multiplication approach. Additionally, we provide a rigorous mathematical analysis demonstrating that orthogonal transformations effectively preserve numerical stability by keeping both the norm and condition number of the matrix unchanged.

4.1 Statistical Analysis

Existing sequential editing methods based on additive update matrices typically take the following form:

$$W = W_0 + \Delta W_1 + \Delta W_2 + \cdots \tag{1}$$

Previous studies have shown that existing methods undermine the numerical stability of the edited parameter matrix, causing significant increases in both the norm and condition number (Ma et al. 2025; Xu et al. 2025). This destabilization negatively impacts the editing performance and general abilities of models. The Frobenius norm of a matrix W can be defined as follows (Kahan 2013):

$$\|W\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |W_{ij}|^2}, \tag{2}$$

where W_{ij} denotes the element in the i-th row and j-th column of W, and $\|W\|_F$ represents the Frobenius norm, which

measures the overall magnitude of the matrix by summing the squares of all its entries. The condition number $\kappa_2(W)$ of a matrix W is (Sun 2000):

$$\kappa_2(W) = \|W\|_2 \cdot \|W^\dagger\|_2, \tag{3}$$

where $\|W\|_2$ denotes the spectral norm of matrix W, which is its largest singular value. $W^\dagger$ represents the Moore-Penrose pseudoinverse of W, and $\|W^\dagger\|_2$ equals the reciprocal of the smallest non-zero singular value of W.

Additive Editing Methods As shown in Figure 2, editing methods such as ROME (Meng et al. 2022) and MEMIT (Meng et al. 2023) cause a substantial increase in both the Frobenius norm and the condition number of the edited matrix, severely undermining its numerical stability. In contrast, methods like RECT (Gu et al. 2024) and AlphaEdit (Fang et al. 2025) demonstrate better capability in limiting edited matrix deviation and, to some extent, preserving its numerical stability. However, even these improved approaches still exhibit noticeable degradation when the number of edits becomes large. This degradation in numerical stability ultimately results in a pronounced decline in both the editing performance and the generalization capabilities of the model across downstream tasks.

Left-Multiplying Orthogonal Matrices In contrast, we sequentially applied random orthogonal transformation by left-multiplying the edited matrix with a series of randomly generated orthogonal matrices at each editing step, thus simulating the sequential editing process:

$$W_i = R_i W_{i-1} \quad \text{for } i = 1, \ldots, k. \tag{4}$$

As depicted in Figure 2, this approach avoids significant growth in matrix norms and condition numbers while rigorously preserving the edited matrix's numerical stability. Through this comparison, it becomes evident that additive update methods in existing knowledge editing frameworks tend to accumulate noise during sequential edits. While effective in early stages, they offer no mechanism to prevent the progressive distortion of the parameter matrix, leading to degradation in numerical properties such as the norm and condition number. In contrast, applying an orthogonal matrix through left multiplication helps preserve these properties, maintaining numerical stability even after a large number of edits, thereby enabling robust knowledge integration.

4.2 Mathematical Analysis

Furthermore, we provide a mathematical analysis showing that, due to the favorable properties of orthogonal transformations, our method is able to strictly preserve the numerical stability of the edited parameter matrix even after a large number of sequential edits. Specifically, orthogonal transformations are capable of keeping key numerical characteristics of matrices unchanged, including the Frobenius norm and the condition number. We begin by recalling the key properties of orthogonal matrices and then proceed with the formal proofs for each of the two quantities. Let $W \in \mathbb{R}^{m\times n}$ be a matrix, and let $R \in \mathbb{R}^{m\times m}$ be an orthogonal matrix:

$$R^\top R = R R^\top = I, \tag{5}$$

where I is the identity matrix. The orthogonal matrix R preserves vector lengths and angles.

Proof of Norm Consider the matrix W' defined as $W' = RW$. We want to prove that the Frobenius norm of W' is equal to that of W:

$$\|W'\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |(RW)_{ij}|^2}. \tag{6}$$

Note that the Frobenius norm squared can be written as the trace of the matrix product:

$$\|W'\|_F^2 = \operatorname{tr}\left((RW)^T (RW)\right). \tag{7}$$

Since R is orthogonal, $R^T R = I$, the identity matrix, thus:

$$\|W'\|_F^2 = \operatorname{tr}\left(W^T I W\right) = \operatorname{tr}\left(W^T W\right). \tag{8}$$

Recognizing that $\operatorname{tr}(W^T W)$ is exactly the squared Frobenius norm of W, we obtain:

$$\|W'\|_F^2 = \|W\|_F^2. \tag{9}$$

Taking the square root on both sides, we conclude:

$$\|W'\|_F = \|W\|_F. \tag{10}$$

This shows that left-multiplying by an orthogonal matrix R does not change the Frobenius norm of W.

Proof of Condition Number Next, we prove that left-multiplying W by an orthogonal matrix does not change its condition number, even when W is not invertible. Since R is orthogonal, the singular values of $W' = RW$ are the same as those of W, implying:

$$\|W'\|_2 = \|RW\|_2 = \|W\|_2. \tag{11}$$

Moreover, by the properties of the Moore-Penrose pseudoinverse, we have:

$$W'^\dagger = (RW)^\dagger = W^\dagger R^\top. \tag{12}$$

Since $R^\top$ is orthogonal, it preserves the spectral norm, so:

$$\|W'^\dagger\|_2 = \|W^\dagger R^\top\|_2 = \|W^\dagger\|_2. \tag{13}$$

Therefore, the condition number defined with the pseudoinverse satisfies:

$$\kappa_2(W') = \|W'\|_2 \cdot \|W'^\dagger\|_2 = \|W\|_2 \cdot \|W^\dagger\|_2 = \kappa_2(W). \tag{14}$$

This shows that left-multiplying by an orthogonal matrix does not change the condition number of W. Therefore, the above proof shows that orthogonal transformation preserve both the Frobenius norm and the condition number of a matrix, thereby strictly ensuring its numerical stability.

5 Method

Existing knowledge editing methods based on additive updates often undermine the numerical stability of the edited matrix, resulting in significant drops in both editing performance and general abilities of the model. To address this issue, we propose a method named **M**ultiplicative **O**rthogonal **S**equential **E**diting (MOSE), which left-multiplies the target parameter matrix with an orthogonal update matrix. MOSE preserves editing performance and generalization by maintaining the numerical stability of the edited matrix, even after extensive sequential edits.

5.1 Orthogonal Transformation-Based Updates

Let $W_0 \in \mathbb{R}^{d\times p}$ represent the original parameter matrix, where $K_0 = \left[k_1^0 \mid k_2^0 \mid \cdots \mid k_{n_0}^0\right] \in \mathbb{R}^{p\times n_0}$ is the matrix containing all the vectors whose representations we want to preserve in a row, and $K_E = [k_1^e \mid k_2^e \mid \cdots \mid k_E^e] \in \mathbb{R}^{p\times n_E}$ is the matrix containing a row of vectors representing the edits we are making in a batch. The target representation of K_E is denoted as $V_E = [v_1^e \mid v_2^e \mid \cdots \mid v_E^e] \in \mathbb{R}^{d\times n_E}$. We introduce an orthogonal transformation matrix $R \in \mathbb{R}^{d\times d}$, which is constrained to satisfy:

$$R^\top R = RR^\top = I_d. \tag{15}$$

This constraint ensures that R is an orthogonal matrix, preserving the geometric structure of the feature space. Similar to MEMIT (Meng et al. 2023), our goal is to minimize the output error for both the knowledge being updated and the knowledge intended to be preserved:

$$\min_W \lambda\|WK_0 - W_0K_0\|_F^2 + \|WK_E - V_E\|_F^2. \tag{16}$$

However, we seek to find an orthogonal matrix that operates directly on the original matrix, rather than computing an additive update matrix. Thus, the optimization problem is:

$$\min_R \lambda\|RW_0K_0 - W_0K_0\|_F^2 + \|RW_0K_E - V_E\|_F^2. \tag{17}$$

Here, $\lambda > 0$ is a regularization parameter that controls the trade-off between preserving the original knowledge and memorizing the new knowledge. The first term ensures that the original knowledge representation remains unchanged, while the second term forces the transformed knowledge representation to match the desired output.

The optimization problem in Eq. (17) is a constrained least-squares problem. When W_0 is fixed, the problem reduces to optimizing R for the best transformation:

$$\min_{R\in O(d)} \|RA - B\|_F^2, \tag{18}$$

where $A = \left[\sqrt{\lambda}W_0K_0 \quad W_0K_E\right] \in \mathbb{R}^{d\times(n_0+n_E)}$ and $B = \left[\sqrt{\lambda}W_0K_0 \quad V_E\right] \in \mathbb{R}^{d\times(n_0+n_E)}$. This is a standard orthogonal Procrustes problem (Schönemann 1966), which has a closed-form solution. The optimal R is obtained via Singular Value Decomposition (SVD) of the matrix $M = BA^\top$:

$$M = U\Sigma V^\top. \tag{19}$$

Thus, the optimal solution for R is:

$$R = UV^\top. \tag{20}$$

By multiplying the obtained orthogonal matrix R with the original parameter matrix W_0, we have successfully completed the knowledge update. Specifically, we employ the method proposed in previous work to derive the components K_0, K_E, and V_E (Meng et al. 2022, 2023), which play crucial roles in the optimization process. For detailed definitions and calculations of these components, the reader is referred to the Appendix.

5.2 Layer Selection Algorithm

Inspired by previous work (Hu et al. 2024; Qiu et al. 2023), we believe that different knowledge resides in different layers of the model, and that selectively modifying specific layers for specific knowledge can enhance editing performance. When editing new knowledge, we first analyze which layer exhibits the strongest activation in response to the specific knowledge. This activation strength reflects how much the output of each layer responds to the given input and is mathematically formulated as:

$$\arg\max_l \left\|\sigma\left(\mathbf{x}\cdot W_{fc}^l\right)\right\|_2, \tag{21}$$

where W_{fc}^l denotes the weight matrix of the feed-forward network (FFN) in the l-th layer of the transformer, and $\sigma(\cdot)$ is the non-linear activation function applied element-wise. This operation captures the transformed representation of the input $\mathbf{x}$ and helps identify the layer most relevant to the knowledge being edited. Considering both how strongly a layer reacts to the target knowledge, we define the final editing layer as follows:

$$l^* = \arg\min_l \left\| \frac{V_E^l - W_0^l K_E^l}{\left\|W_0^l\right\|_2 \cdot \sigma(\mathbf{x}\cdot W_{fc}^l)} \right\|_F, \tag{22}$$

where the denominator $\|W_0^l\|_2$ acts as a normalization term, facilitating fair comparison across layers. Ultimately, we identify the editing target as $W_0^{l^*}$ at layer l^*. In addition, we found that editing multiple layers simultaneously can further enhance editing performance. Therefore, once we identify the target layer for editing, we modify that layer along with its two adjacent layers. In later sections, we present corresponding ablation experiments and provide further analysis to support this proposed approach.

6 Experiments

6.1 Experimental Setup

Experiments were conducted on three LLMs: LLaMA3-8B (Dubey, Jauhri, and et al. 2024), LLaMA2-13B (Touvron, Martin, and et al. 2023), and Qwen2.5-7B (Yang et al. 2024a). The baseline editing methods used were ROME (Meng et al. 2022), MEMIT (Meng et al. 2023), RECT (Gu et al. 2024), EMMET (Gupta, Baskaran, and Anumanchipalli 2024), PRUNE (Ma et al. 2025) and AlphaEdit (Fang et al. 2025). The editing performance was evaluated using two types of datasets: factual knowledge-based datasets, including ZsRE (Levy et al. 2017) and CounterFact (Meng et al. 2022), and a conceptual knowledge-based dataset, ConceptEdit (Wang et al. 2024b). The models were assessed based on metrics such as reliability, generalization, and locality (Meng et al. 2022, 2023; Yao et al. 2023; Xu et al. 2025). To measure the general abilities of the models before and after editing, four downstream tasks were selected: **natural language inference** (Dagan, Glickman, and Magnini 2005), **summarization** (Gliwa et al. 2019), **open-domain question-answering** (Kwiatkowski et al. 2019), and **sentiment analysis** (Socher et al. 2013). Due to page

Method	Model	CounterFact			ConceptEdit-Inter		
		Reliability	Generalization	Locality	Reliability	Generalization	Locality
ROME	LLama3-8B	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MEMIT		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RECT		0.5688	0.3288	0.2517	0.3531	0.2084	0.1502
EMMET		0.6398	0.4937	0.3253	0.3877	0.2258	0.1724
PRUNE		0.7886	0.7140	0.5763	0.6295	0.4414	0.3193
AlphaEdit		0.9018	0.8260	0.7831	0.7012	0.6007	0.5212
MOSE		**0.9887**	**0.9863**	**0.8972**	**0.7859**	**0.7275**	**0.6856**
ROME	Qwen2.5-7B	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MEMIT		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RECT		0.6203	0.4745	0.3582	0.3737	0.2306	0.1738
EMMET		0.6702	0.5589	0.4771	0.4593	0.2641	0.1903
PRUNE		0.8115	0.7860	0.6823	0.6708	0.5009	0.4120
AlphaEdit		0.9519	0.9241	0.8418	0.7346	0.6453	0.6116
MOSE		**0.9981**	**0.9902**	**0.9098**	**0.8012**	**0.7547**	**0.7069**

Table 1: In the single-sequential editing scenario, the editing performance of different methods on CounterFact and ConceptEdit-Inter. We performed 4000 sequential edits.

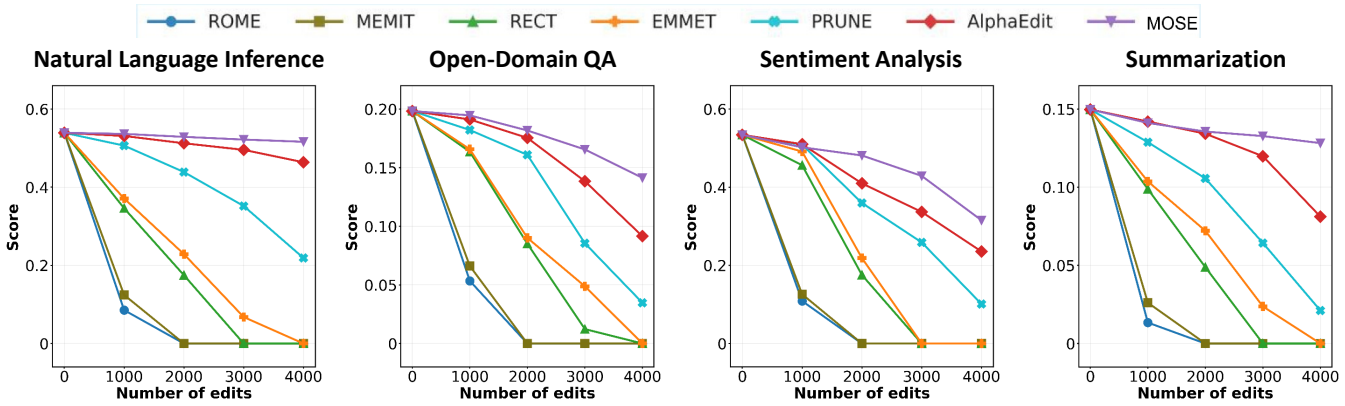

Figure 3: Edited on the CounterFact dataset, the general task performance of varying methods with LLaMA3-8B as the number of edits increases, under the single-sequential scenario.

limitations, the experiments presented in our main text include only a representative subset of the results. Readers can refer to the Appendix for further details on the experimental setups and more comprehensive results.

6.2 Results in Single-Sequential Editing Scenarios

This section illustrates, in the single-sequential editing scenario, the editing performance and general abilities across downstream tasks of the edited model.

Editing Performance We aim for the sequentially edited model to retain all previously edited knowledge. To evaluate this, we performed a sequence of edits on the model and assessed its editing performance on the edited knowledge, which serves as an indicator of the model's retention ability. Table 1 reports the editing performance of LLaMA3-8B and Qwen2.5-7B after 4000 sequential edits using both the CounterFact and Concept-Inter datasets across different editing methods. We observe that under previous editing methods, the model's performance deteriorates significantly as the number of sequential edits increases, regardless of the type of knowledge being edited. This suggests that these methods may introduce substantial disruptions to the model, reducing its ability to preserve previously integrated knowledge. In contrast, MOSE demonstrates stronger retention performance in single-sequence editing scenarios by effectively maintaining the numerical stability of the model's parameter matrix throughout the editing process.

General Abilities Applying CounterFact as the editing dataset, Figure 3 shows the task performance of different editing methods on LLaMA3-8B. It can be observed that when sequential edits are performed using traditional methods, the general abilities of the edited model fluctuate significantly and tend to decline as the number of edits increases. This issue becomes especially pronounced when the number of edits is large, severely limiting the model's scalability. In contrast, by minimizing deviations between the pre- and post-edited models, MOSE effectively preserves the general abilities of the edited model across downstream tasks.

Method	Model	CounterFact			ConceptEdit-Inter		
		Reliability	Generalization	Locality	Reliability	Generalization	Locality
ROME	LLama3-8B	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MEMIT		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RECT		0.5266	0.3075	0.2382	0.3234	0.1993	0.1397
EMMET		0.6287	0.4695	0.3114	0.3866	0.2178	0.1563
PRUNE		0.7738	0.6899	0.5190	0.5682	0.4097	0.3083
AlphaEdit		0.8222	0.7835	0.7091	0.6981	0.5928	0.4977
MOSE		**0.9422**	**0.9361**	**0.8819**	**0.7585**	**0.7191**	**0.6475**
ROME	Qwen2.5-7B	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MEMIT		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RECT		0.6080	0.4679	0.3239	0.3659	0.2266	0.1593
EMMET		0.6579	0.5125	0.4338	0.4402	0.2461	0.1751
PRUNE		0.7707	0.7527	0.6300	0.6395	0.4800	0.3857
AlphaEdit		0.9427	0.8911	0.8094	0.6733	0.6162	0.5743
MOSE		**0.9775**	**0.9514**	**0.9031**	**0.7526**	**0.7258**	**0.6731**

Table 2: In the batch-sequential editing scenario, the editing performance of different methods on CounterFact and ConceptEdit-Inter. We set the batch size to 10 and performed 500 sequential edits.

6.3 Results in Batch-Sequential Editing Scenarios

To further demonstrate the superiority and robustness of our method, we introduce a more challenging scenario—*batch-sequential editing*—which requires modifying multiple pieces of knowledge during each editing step. This section comprehensively illustrates the editing performance of the edited models. Using COUNTERFACT and CONCEPTEDIT-INTER as the editing datasets, Table 2 presents a comprehensive comparative evaluation of the editing performance of LLAMA3-8B and QWEN2.5-7B using different methods under the batch-sequential setting. In these experiments, we set the batch size to 10 and performed 500 editing steps, evaluating each model's ability to retain all previously edited knowledge. The results show that, under prior editing methods, simultaneously editing multiple knowledge pieces at each step causes more severe degradation than editing one piece at a time, leading to diminished overall performance and revealing that previous knowledge-editing methods face greater challenges in this scenario. In contrast, MOSE consistently maintains strong editing performance under batch-sequential editing and demonstrates superior scalability compared with prior approaches. More experiments of general abilities can refer to the Appendix.

6.4 Ablation Study

To evaluate the effectiveness of the layer selection algorithm in MOSE, we conducted a series of ablation studies under three experimental setups. In the first setup, editing is performed on a predefined layer, where the number of layer matches it used in the ROME baseline. In the second setup, editing is applied to a single layer that is selected using our layer selection algorithm. In the third setup, which corresponds to the complete MOSE, editing is performed on the selected layer as well as its immediate neighboring layers. As shown in Figure 4, editing a predefined layer yields the weakest performance, indicating that knowledge is distributed across layers rather than confined to a single one. Thus, fixed-layer strategies fall short in handling diverse knowledge. Selecting a single layer via a layer selection algorithm improves results, but MOSE, which edits both the selected and neighboring layers, performs best in batch-sequential settings.

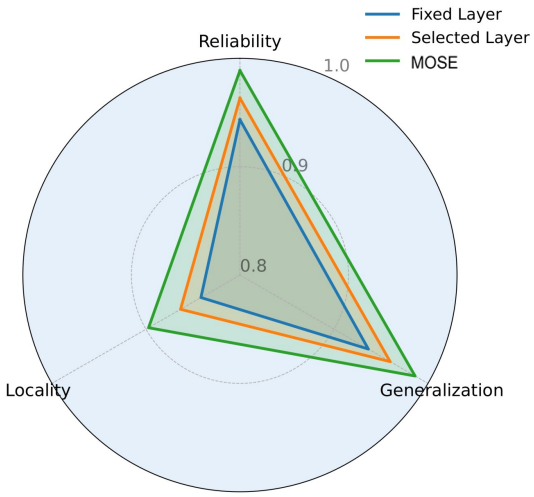

Figure 4: Ablation analysis of editing performance in the batch-sequential scenario. The experiment was conducted with LLaMA3-8B on CounterFact dataset.

7 Conclusion

This paper addresses sequential model editing and shows that existing additive update methods cause deviations and numerical instability in the parameter matrix. Through statistical and mathematical analysis, we demonstrate that orthogonal transformations preserve numerical properties during editing. Based on this, we propose MOSE, which maintains stability by multiplicatively applying orthogonal transformations to the original matrix. Experiments confirm that MOSE effectively maintains numerical stability, thereby preserving both editing performance and general abilities.

Acknowledgements

This work is partially funded by the National Science and Technology Major Project (No.2023ZD0121103). We would like to express gratitude to the anonymous reviewers for their kind comments.

References

Cao, N. D.; Aziz, W.; and Titov, I. 2021. Editing Factual Knowledge in Language Models. In Moens, M.; Huang, X.; Specia, L.; and Yih, S. W., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, 6491–6506. Association for Computational Linguistics.

Dagan, I.; Glickman, O.; and Magnini, B. 2005. The PASCAL Recognising Textual Entailment Challenge. In Candela, J. Q.; Dagan, I.; Magnini, B.; and d'Alché-Buc, F., eds., *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*, 177–190. Springer.

Dong, Q.; Dai, D.; Song, Y.; Xu, J.; Sui, Z.; and Li, L. 2022. Calibrating Factual Knowledge in Pretrained Language Models. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 5937–5947. Association for Computational Linguistics.

Dubey, A.; Jauhri, A.; and et al. 2024. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783.

Fang, J.; Jiang, H.; Wang, K.; Ma, Y.; Shi, J.; Wang, X.; He, X.; and Chua, T. 2025. AlphaEdit: Null-Space Constrained Knowledge Editing for Language Models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Gliwa, B.; Mochol, I.; Biesek, M.; and Wawer, A. 2019. SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization. In Wang, L.; Cheung, J. C. K.; Carenini, G.; and Liu, F., eds., *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 70–79. Hong Kong, China: Association for Computational Linguistics.

Gu, J.; Xu, H.; Ma, J.; Lu, P.; Ling, Z.; Chang, K.; and Peng, N. 2024. Model Editing Harms General Abilities of Large Language Models: Regularization to the Rescue. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, 16801–16819. Association for Computational Linguistics.

Gupta, A.; Baskaran, S.; and Anumanchipalli, G. 2024. Rebuilding ROME : Resolving Model Collapse during Sequential Model Editing. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, 21738–21744. Association for Computational Linguistics.

Hartvigsen, T.; Sankaranarayanan, S.; Palangi, H.; Kim, Y.; and Ghassemi, M. 2023. Aging with GRACE: Lifelong Model Editing with Discrete Key-Value Adaptors. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *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*.

Hu, C.; Cao, P.; Chen, Y.; Liu, K.; and Zhao, J. 2024. WilKE: Wise-Layer Knowledge Editor for Lifelong Knowledge Editing. In Ku, L.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, 3476–3503. Association for Computational Linguistics.

Huang, Z.; Shen, Y.; Zhang, X.; Zhou, J.; Rong, W.; and Xiong, Z. 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.

Kahan, W. 2013. A tutorial overview of vector and matrix norms. *University of California, Berkeley, CA, USA, Lecture notes*, 19.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A. P.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; Toutanova, K.; Jones, L.; Kelcey, M.; Chang, M.; Dai, A. M.; Uszkoreit, J.; Le, Q.; and Petrov, S. 2019. Natural Questions: a Benchmark for Question Answering Research. *Trans. Assoc. Comput. Linguistics*, 7: 452–466.

Levy, O.; Seo, M.; Choi, E.; and Zettlemoyer, L. 2017. Zero-Shot Relation Extraction via Reading Comprehension. In Levy, R.; and Specia, L., eds., *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, 333–342. Association for Computational Linguistics.

Liu, W.; Lin, R.; Liu, Z.; Liu, L.; Yu, Z.; Dai, B.; and Song, L. 2018. Learning towards Minimum Hyperspherical Energy. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 6225–6236.

Ma, J.; Ling, Z.; Zhang, N.; and Gu, J. 2024. Neighboring Perturbations of Knowledge Editing on Large Language Models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Ma, J.; Wang, H.; Xu, H.; Ling, Z.; and Gu, J. 2025. Perturbation-Restrained Sequential Model Editing. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2022.

Locating and Editing Factual Associations in GPT. In *NeurIPS*.

Meng, K.; Sharma, A. S.; Andonian, A. J.; Belinkov, Y.; and Bau, D. 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.

Mitchell, E.; Lin, C.; Bosselut, A.; Finn, C.; and Manning, C. D. 2022a. Fast Model Editing at Scale. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Mitchell, E.; Lin, C.; Bosselut, A.; Manning, C. D.; and Finn, C. 2022b. Memory-Based Model Editing at Scale. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 15817–15831. PMLR.

Qiu, Z.; Liu, W.; Feng, H.; Xue, Y.; Feng, Y.; Liu, Z.; Zhang, D.; Weller, A.; and Schölkopf, B. 2023. Controlling Text-to-Image Diffusion by Orthogonal Finetuning. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *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*.

Schönemann, P. H. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1): 1–10.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 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*, 1631–1642. ACL.

Sun, J. 2000. Condition Number and Backward Error for the Generalized Singular Value Decomposition. *SIAM J. Matrix Anal. Appl.*, 22(2): 323–341.

Touvron, H.; Martin, L.; and et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *CoRR*, abs/2307.09288.

Wang, P.; Li, Z.; Zhang, N.; Xu, Z.; Yao, Y.; Jiang, Y.; Xie, P.; Huang, F.; and Chen, H. 2024a. WISE: Rethinking the Knowledge Memory for Lifelong Model Editing of Large Language Models. In Globersons, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Wang, X.; Mao, S.; Deng, S.; Yao, Y.; Shen, Y.; Liang, L.; Gu, J.; Chen, H.; and Zhang, N. 2024b. Editing Conceptual Knowledge for Large Language Models. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, 706–724. Association for Computational Linguistics.

Xu, H.; Ma, J.; Ling, Z.; Zhang, N.; and Gu, J. 2025. Constraining Sequential Model Editing with Editing Anchor Compression. In Chiruzzo, L.; Ritter, A.; and Wang, L., eds., *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, 5499–5515. Association for Computational Linguistics.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2024a. Qwen2.5 Technical Report. *CoRR*, abs/2412.15115.

Yang, W.; Sun, F.; Tan, J.; Ma, X.; Su, D.; Yin, D.; and Shen, H. 2024b. The Fall of ROME: Understanding the Collapse of LLMs in Model Editing. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, 4079–4087. Association for Computational Linguistics.

Yao, Y.; Wang, P.; Tian, B.; Cheng, S.; Li, Z.; Deng, S.; Chen, H.; and Zhang, N. 2023. Editing Large Language Models: Problems, Methods, and Opportunities. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, 10222–10240. Association for Computational Linguistics.

Zhang, N.; Tian, B.; Cheng, S.; Liang, X.; Hu, Y.; Xue, K.; Gou, Y.; Chen, X.; and Chen, H. 2024. InstructEdit: Instruction-Based Knowledge Editing for Large Language Models. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, 6633–6641. ijcai.org.

Zhang, Y.; Li, Y.; Cui, L.; Cai, D.; Liu, L.; Fu, T.; Huang, X.; Zhao, E.; Zhang, Y.; Chen, Y.; Wang, L.; Luu, A. T.; Bi, W.; Shi, F.; and Shi, S. 2023. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *CoRR*, abs/2309.01219.

Zheng, C.; Li, L.; Dong, Q.; Fan, Y.; Wu, Z.; Xu, J.; and Chang, B. 2023. Can We Edit Factual Knowledge by In-Context Learning? In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, 4862–4876. Association for Computational Linguistics.