FACTUAL ASSOCIATION EDITING WITHOUT PRETRAINED KNOWLEDGE REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a closed-form factual association editing method for language models that completely eliminates the need to precompute and preserve pretrained knowledge representations. Unlike existing approaches that rely on proxy representations of pretrained knowledge obtained from sampled datasets, the proposed method formulates factual association editing as a well-conditioned optimization problem that enables stable and efficient editing using only new knowledge. During the editing process, we apply a principled treatment of ill-conditioning via within-representation centering and truncated singular value decomposition to prevent unstable or excessively amplified updates. The experimental results show that the proposed approach achieves editing performance comparable to that of state-of-the-art methods that rely on pretrained knowledge representations, while reducing amortized runtime for editing by up to 30×, i.e., from 188 to 6 minutes, and requiring zero storage for pretrained knowledge representations. Even in large-scale editing (e.g., 2,000 edits), the proposed method maintains competitive editing performance, exhibiting only a 0.56% degradation on GPT-J without using any pretrained knowledge representations at all. These results provide empirical evidence that competitive factual association editing can be achieved without constructing an external proxy for the knowledge to be preserved. The code is available on an anonymous repository[1].

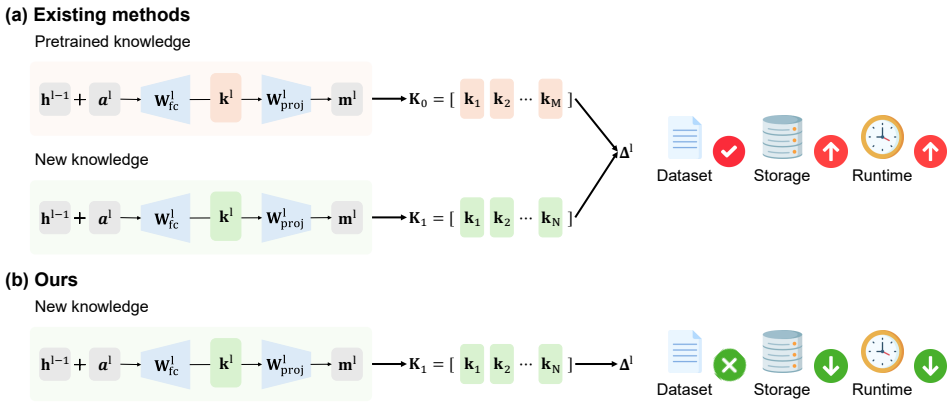

Figure 1: Comparison of existing closed-form methods and the proposed method (Ours). Existing methods require both the pretrained knowledge represented by $\mathbf{K}_0$ and the new knowledge represented by $\mathbf{K}_1$ for factual association editing, whereas the proposed method requires only $\mathbf{K}_1$.

1 INTRODUCTION

Language models (Radford et al., 2019; Chang et al., 2024) have demonstrated a remarkable ability to store and recall factual knowledge acquired during pretraining, effectively functioning as implicit knowledge bases. This capability enables a wide range of applications, including question answering (Rajpurkar et al., 2016; Brown et al., 2020; Lewis et al., 2020), reasoning (Wei et al., 2022;

[1] https://anonymous.4open.science/r/Anonymous-FCB2

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Kojima et al., 2022), and information retrieval (Karpukhin et al., 2020; Guu et al., 2020; Guo et al., 2025). However, despite their strong memorization capacity, language models remain fundamentally limited in their ability to update or revise stored knowledge after training. As a result, over time, they may retain outdated, incorrect, or sensitive information that users wish to modify.

To address this limitation, knowledge editing (Sinitsin et al., 2020; Mitchell et al., 2022a; Meng et al., 2022; 2023) has been proposed to update the language model's specific knowledge without retraining the model from scratch. Among these, weight parameter-update approaches (Meng et al., 2022; 2023; Fang et al., 2025) follow a locate-then-edit paradigm, in which the model first identifies weight parameters associated with a specific target fact and then applies localized updates to modify that knowledge. In contrast, alternative approaches (Mitchell et al., 2022b) avoid direct parameter modification by leveraging external memory or retrieval mechanisms to override model outputs at inference time. While the latter methods offer different trade-offs between efficiency and modularity, locate-then-edit approaches remain attractive due to their ability to directly integrate new knowledge into localized model weights in a closed-form manner without auxiliary memory systems.

A key challenge in closed-form-based locate-then-edit methods lies in preserving pretrained knowledge while incorporating new information. The weight updates introduced to encode new knowledge may inadvertently overwrite existing pretrained knowledge not accounted for, degrading previously learned information. In practice, existing approaches typically use representations of pretrained knowledge to constrain such unintended changes. However, fully characterizing and storing pretrained knowledge is impractical due to its large scale and highly distributed nature. Consequently, existing approaches (Meng et al., 2022; 2023; Fang et al., 2025) approximate these representations of pretrained knowledge from sampled data, such as Wikipedia (Vrandecić & Krötzsch, 2014), introducing a dependency on proxy knowledge. This fundamentally limits the scalability and practicality of knowledge editing, as it requires storing large representations of pretrained knowledge and incorporating them into every editing operation, incurring substantial computation and memory cost.

In this work, we introduce a novel method for closed-form factual association editing that uses only newly introduced knowledge ($\mathbf{K}_1$), without requiring pretrained knowledge representations ($\mathbf{K}_0$), as illustrated in Fig. 1. In contrast to existing approaches (Meng et al., 2022; 2023; Fang et al., 2025) that rely on retaining such pretrained knowledge representations ($\mathbf{K}_0$), the proposed method enables efficient and stable knowledge editing without precomputing and storing $\mathbf{K}_0$. This approach substantially improves the scalability of factual association editing by eliminating precomputation and storage requirements for pretrained knowledge while maintaining high editing fidelity. Moreover, the proposed method admits a closed-form solution and requires neither additional modules nor auxiliary models, making it a simple and practical method for factual association editing. Tab. 1 summarizes the differences between the proposed method (Ours) and existing knowledge editing methods.

Table 1: Comparison of representative knowledge editing methods. Closed-form indicates whether parameter updates are computed analytically. Precomp. indicates whether pretrained knowledge representations are required for editing. Add. Module indicates whether additional modules, such as adapters, routers, and side memories, are integrated into the base model. Aux. Model indicates whether a separately trained editor, hypernetwork, or counterfactual model is used.

Method	Closed-form	No Precomp.	No Add. Module	No Aux. Model
MEMIT	✓	×	✓	✓
AlphaEdit	✓	×	✓	✓
FastMEMIT	✓	×	✓	✓
MEND	×	✓	✓	×
SERAC	×	✓	✓	×
GRACE	×	✓	×	✓
MELO	×	✓	×	✓
WISE	×	✓	×	✓
Ours	✓	✓	✓	✓

In particular, we pose the following research question: *Can closed-form factual association editing remain effective and stable without access to pretrained knowledge representations?* Our analysis reveals that removing pretrained knowledge representations can render the underlying closed-form system ill-conditioned, resulting in a substantial degradation in editing performance. To address this challenge, we formulate factual association editing without access to pretrained knowledge representations as a well-conditioned optimization problem. Our formulation regularizes directions associated with near-zero singular values, preventing the unstable or excessively amplified updates that would otherwise arise from ill-conditioning. Consequently, the proposed method enables stable and efficient factual association editing using only representations of newly introduced knowledge.

Our experimental results show that the proposed method enables closed-form factual association editing without pretrained knowledge representations while substantially improving editing efficiency over state-of-the-art approaches that rely on such representations (Meng et al., 2023; Ma et al., 2025; Gu et al., 2024; Fang et al., 2025). For instance, without requiring 5–20 hours of precomputation and additional 0.76–6.00 GiB of storage required for the evaluated existing baselines, the proposed method achieves up to a 30× runtime speedup for editing and improves the score on the MCF dataset on LLaMA 3 (Grattafiori et al., 2024) by 2.92%, while limiting performance degradation to 0.56% and 1.41% on GPT-J (Wang & Komatsuzaki, 2021) and GPT-2 XL (Radford et al., 2019), respectively. These findings demonstrate that efficient and scalable factual association editing is feasible without pretrained knowledge representations, implying that this approach can be a promising direction for further research.

2 Factual Association Editing

2.1 Definition and Problem

In language models (Radford et al., 2019; Chang et al., 2024), knowledge can be represented and retrieved as factual associations in the form of triples, comprising a subject s, a relation r, and an object o. Given a prompt $p_i = p(s_i, r_i)$ constructed from a subject–relation pair, the model predicts the corresponding object o_i. For example, in the sentence "*The capital of Norway is Oslo*," the subject s corresponds to '*Norway*', the relation r corresponds to '*capital*', and the object o corresponds to '*Oslo*'. Basically, a set of factual associations, as a form of knowledge, is acquired through training a model on datasets, meaning that they are inherently dependent on training data and are difficult to modify without updating the model.

Given a prompt for a subject–relation pair, $p_i = p(s_i, r_i)$, the objective of factual association editing is to modify the predicted object from o_i to a new target object o_k such that $o_i \neq o_k$. We represent the set of factual associations as follows:

$$\mathcal{E} = \{(s_i, r_i, o_i) \mid i\} \quad \text{s.t.} \quad \forall i, j.\ (s_i = s_j) \wedge (r_i = r_j) \Rightarrow (o_i = o_j). \tag{1}$$

For example, the above sentence can be edited to "*The capital of Norway is Rome*," changing o from *Oslo* to *Rome*, while s and r remain unchanged.

Since locate-then-edit approaches (Meng et al., 2022; 2023; Fang et al., 2025) locate and modify the model's weight parameters to associate a given prompt $p_i = p(s_i, r_i)$ with a new target object o_j, it is necessary to preserve a dataset that helps the model continue to produce the original objects o_k for prompts $p_k = p(s_k, r_k)$ for all $k \neq i$. In practice, hidden representations of the model learned from a subset of data are typically preserved and repeatedly utilized throughout the editing process to maintain unedited factual associations.

2.2 Locating Factual Associations

Auto-regressive language models (Vaswani et al., 2017; Radford et al., 2018; 2019) generate output tokens sequentially, where the l-th layer's hidden state representation $\mathbf{h}^l$ is computed as follows:

$$\mathbf{h}^l = \mathbf{h}^{l-1} + \mathbf{a}^l + \mathbf{m}^l. \tag{2}$$

where $\mathbf{a}^l$ is the output of the attention module, and $\mathbf{m}^l$ is the output of the feed-forward module.

Following prior work on locate-then-edit methods (Meng et al., 2022), we focus on updating the feed-forward module output $\mathbf{m}^l$ in Eq. (2) for factual association editing, which is defined as follows:

$$\mathbf{m}^l = \mathbf{W}^l_{\text{proj}} \sigma(\mathbf{W}^l_{\text{fc}} \gamma(\mathbf{h}^{l-1} + \mathbf{a}^l)). \tag{3}$$

where $\mathbf{W}^l_{\text{proj}}$ and $\mathbf{W}^l_{\text{fc}}$ are the weight matrices, σ is the non-linear activation function, and γ is the layer normalization. Specifically, we update $\mathbf{W}^l_{\text{proj}}$ in Eq. (3) at the middle layers of the model for factual association editing, which have been identified by causal tracing (Meng et al., 2022) as the most critical and effective part for editing.

3 EDITING WITHOUT PRETRAINED KNOWLEDGE REPRESENTATIONS

3.1 ELIMINATING PRETRAINED KNOWLEDGE REPRESENTATIONS

To preserve pretrained knowledge not subject to modification during factual association editing, prior approaches (Meng et al., 2022; 2023; Fang et al., 2025) explicitly include the pretrained knowledge pair, denoted as $\{\mathbf{K}_0, \mathbf{V}_0\}$, to be maintained in their objective function alongside the new knowledge pair $\{\mathbf{K}_1, \mathbf{V}_1\}$, as follows:

$$\mathbf{\Delta} = \arg\min_{\tilde{\mathbf{\Delta}}} \left(\left\| (\mathbf{W} + \tilde{\mathbf{\Delta}})\mathbf{K}_1 - \mathbf{V}_1 \right\|^2 + \left\| (\mathbf{W} + \tilde{\mathbf{\Delta}})\mathbf{K}_0 - \mathbf{V}_0 \right\|^2 \right). \tag{4}$$

Here, $\mathbf{W} = \mathbf{W}_{\text{proj}}^l$ for notational simplicity, $\mathbf{\Delta}$ is the weight perturbation induced by factual association editing, i.e., we update $\mathbf{W}$ to $\mathbf{W}+\mathbf{\Delta}$, $\mathbf{K} = [\mathbf{k}_1^l \mid \mathbf{k}_2^l| \cdots |\mathbf{k}_n^l]$, where $\mathbf{k}_i^l = \sigma(\mathbf{W}_{\text{fc}}^l \gamma(\mathbf{h}_i^{l-1} + \mathbf{a}_i^l))$ denotes the key representation vector for the i-th knowledge instance, where $\mathbf{V} = [\mathbf{v}_1^l | \mathbf{v}_2^l | \cdots | \mathbf{v}_n^l]$, with $\mathbf{v}_i^l = \mathbf{m}_i^l$ being the i-th memory vector corresponding to $\mathbf{k}_i^l$ in Eq. (3).

The goal of Eq. (4) is to find $\mathbf{\Delta}$ such that both $\mathbf{V}_0$ and $\mathbf{V}_1$ are maintained after factual association editing, given their respective key representations $\mathbf{K}_0$ and $\mathbf{K}_1$ over $\mathbf{W} + \mathbf{\Delta}$, i.e., achieving $\mathbf{V}_0 \approx (\mathbf{W} + \mathbf{\Delta})\mathbf{K}_0$ and $\mathbf{V}_1 \approx (\mathbf{W} + \mathbf{\Delta})\mathbf{K}_1$.

Assuming $\mathbf{W}\mathbf{K}_0\mathbf{K}_0^\top = \mathbf{V}_0\mathbf{K}_0^\top$ by following prior work (Meng et al., 2023), we can derive the corresponding closed-form solution to Eq. (4) as:

$$\mathbf{\Delta} = (\mathbf{V}_1 - \mathbf{W}\mathbf{K}_1)\mathbf{K}_1^\top (\mathbf{K}_0\mathbf{K}_0^\top + \mathbf{K}_1\mathbf{K}_1^\top)^{-1}. \tag{5}$$

To solve Eq. (5), the pretrained key representations $\mathbf{K}_0$ must be explicitly precomputed and preserved for editing, which poses challenges for scalable editing.

Our goal is to enable factual association editing without incorporating the pretrained knowledge representations $\mathbf{K}_0$ into Eq. (4) and (5), which improves scalability of editing by avoiding high memory and computational costs incurred by $\mathbf{K}_0$. To enable editing without $\mathbf{K}_0$, we redefine Eq. (4) by removing the second term involving both $\mathbf{K}_0$ and $\mathbf{V}_0$, yielding:

$$\mathbf{\Delta} = \arg\min_{\tilde{\mathbf{\Delta}}} \|(\mathbf{W} + \tilde{\mathbf{\Delta}})\mathbf{K}_1 - \mathbf{V}_1\|^2. \tag{6}$$

By analyzing $\mathbf{W}$ as a linear associative memory, similar to existing works (Meng et al., 2023; Fang et al., 2025), Eq. (6) can be optimized by solving the following normal equation, yielding the closed-form solution:

$$(\mathbf{W} + \mathbf{\Delta})\mathbf{K}_1\mathbf{K}_1^\top = \mathbf{V}_1\mathbf{K}_1^\top. \tag{7}$$

$$\mathbf{\Delta} = (\mathbf{V}_1 - \mathbf{W}\mathbf{K}_1)\mathbf{K}_1^\top (\mathbf{K}_1\mathbf{K}_1^\top)^\dagger, \tag{8}$$

where † denotes the Moore-Penrose pseudoinverse (Moore, 1920).

3.2 IMPROVING NUMERICAL CONDITIONING

Although Eq. (8) admits a closed-form solution as a linear equation via singular value decomposition (SVD), naively solving it is likely to substantially amplify the relative error in $\mathbf{\Delta}$. In particular, it can introduce large perturbations to the current weight $\mathbf{W}$ through $\mathbf{\Delta}$ (i.e., $\mathbf{W} + \mathbf{\Delta}$), potentially causing the model to forget the pretrained knowledge represented by $\mathbf{K}_0$, which is not explicitly considered in our objective (Eq. (6)). To address this, we aim to reduce the upper bound on error amplification in $\mathbf{\Delta}$ while improving the stability of the solution to Eq. (8).

One way to reduce error amplification in $\mathbf{\Delta}$ is to minimize the condition number of the singular value matrix $\mathbf{S} = \text{diag}(\sigma_1, \ldots, \sigma_r)$ of $\mathbf{K}_1$, where $\mathbf{K}_1\mathbf{K}_1^\top = \mathbf{U}\mathbf{S}^2\mathbf{U}^\top$, when solving Eq. (8). In general, given $\mathbf{S}$, the condition number is defined as $\kappa(\mathbf{S}) = \frac{\sigma_{\max}(\mathbf{S})}{\sigma_{\min}(\mathbf{S})}$ where $\sigma_{\max}(\mathbf{S})$ and $\sigma_{\min}(\mathbf{S})$ denote the largest and smallest non-zero singular values, respectively. Since the condition number κ quantifies how sensitive the output is to small perturbations in the input, a large κ reflects that even minor errors in $\mathbf{V}_1 - \mathbf{W}\mathbf{K}_1$ in Eq. (8) can be substantially amplified in $\mathbf{\Delta}$.

To avoid squaring the condition number in Eq. (8), we use the identity $\mathbf{K}_1^\top(\mathbf{K}_1\mathbf{K}_1^\top)^\dagger = \mathbf{K}_1^\dagger$ as follows:

$$\mathbf{\Delta} = (\mathbf{V}_1 - \mathbf{W}\mathbf{K}_1)\mathbf{K}_1^\dagger. \tag{9}$$

where $\mathbf{K}_1^\dagger$ denotes the pseudoinverse of $\mathbf{K}_1$. Considering non-zero singular values, the condition number of $\mathbf{K}_1\mathbf{K}_1^\top$ is the square of the condition number of $\mathbf{K}_1$, as in $\kappa(\mathbf{K}_1\mathbf{K}_1^\top) = \kappa(\mathbf{K}_1)^2$. Thus, directly computing the pseudoinverse of $\mathbf{K}_1$, as in Eq. (9), avoids the squared condition number arising from $\mathbf{K}_1\mathbf{K}_1^\top$, with $\kappa(\mathbf{K}_1) = \frac{\sigma_{\max}(\mathbf{K}_1)}{\sigma_{\min}(\mathbf{K}_1)}$. This formulation therefore improves numerical stability compared with the formulation in Eq. (8).

To further minimize the condition number κ, we employ truncated singular value decomposition (TSVD) (Hansen, 1987), which discards small singular values that may inflate the condition number κ. Applying TSVD also alleviates the problem that solving Eq. (9) using only $\mathbf{K}_1$ without $\mathbf{K}_0$ may lead to an ill-conditioned system. A rank-deficient matrix has singular values that are theoretically zero, but due to finite-precision arithmetic during SVD, they may appear as very small nonzero values. By truncating them along with other small singular values based on a specific criterion, we mitigate the influence of these small components and improve the conditioning of the pseudoinverse computation. Therefore, TSVD serves as an effective regularization for solving Eq. (9).

3.3 Removing the All-Ones Component

To further improve the effectiveness of factual association editing without the pretrained knowledge representations $\mathbf{K}_0$, we apply within-representation centering to the new knowledge representations $\mathbf{K}_1 \in \mathbb{R}^{d\times N}$ in Eq. (9), where each column corresponds to a key representation for an edit.

Specifically, we define the centered key matrix $\widetilde{\mathbf{K}}_1$, which removes the mean component in the all-ones direction from each key representation, as:

$$\widetilde{\mathbf{K}}_1 = \Big(\mathbf{I} - \frac{1}{d}\mathbf{1}\mathbf{1}^\top\Big)\mathbf{K}_1. \tag{10}$$

where $\mathbf{I} \in \mathbb{R}^{d\times d}$ is the identity matrix, $\mathbf{1}\mathbf{1}^\top \in \mathbb{R}^{d\times d}$ denotes the all-ones matrix, and $\mathbf{I} - \frac{1}{d}\mathbf{1}\mathbf{1}^\top$ is the centering matrix.

By replacing $\mathbf{K}_1^\dagger$ in Eq. (9) with the pseudoinverse of the centered key matrix $\widetilde{\mathbf{K}}_1$ in Eq. (10), we reformulate Eq. (9) as:

$$\mathbf{\Delta} = (\mathbf{V}_1 - \mathbf{W}\mathbf{K}_1)\widetilde{\mathbf{K}}_1^\dagger. \tag{11}$$

We focus on the anisotropic geometry of language model representations, which are often dominated by common directions (Mu et al., 2018; Ethayarajh, 2019; Biś et al., 2021; Wang et al., 2021). Previous work (Wang et al., 2021) has also shown that the average representation in language models is nearly aligned with the first principal component of the representations whose columns are representation vectors. In their PCA, each representation is shifted to have a mean of zero across its dimensions, which is equivalent to within-representation centering. For the centered key matrix $\widetilde{\mathbf{K}}_1$, the first principal component direction corresponds to the dominant left singular-vector direction associated with the largest singular value. We remove the first principal component only for sequential editing. The ablation study in Appendix C demonstrates that centering improves both efficacy and generalization.

3.4 Determining the Retained Rank

To determine the number of singular values retained after TSVD of S in $\widetilde{\mathbf{K}}_1 = \mathbf{U}\mathbf{S}\mathbf{V}^\top$, we introduce the spectral entropy criterion and the cumulative $\mathbf{S}^4$ criterion. The spectral entropy criterion uses the effective rank, which provides a continuous measure of the dimensionality of the singular value distribution (Roy & Vetterli, 2007). Let $\mathbf{S} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$. The spectral entropy is defined as $H(\mathbf{S}) = -\sum_{i=1}^{r} p_i \log p_i$, where $p_i = \frac{\sigma_i}{\sum_{j=1}^{r}\sigma_j}$. The effective rank is then computed as $\mathrm{erank}(\mathbf{S}) = \exp(H(\mathbf{S}))$. Since the effective rank is generally non-integer, we set the retained rank to $k_{\mathrm{erank}} = \lceil \mathrm{erank}(\mathbf{S}) \rceil$ and retain the k_{erank} largest singular values.

Alternatively, the retained rank can be determined by raising S to the fourth power, i.e., $\mathbf{S}^4$. The cumulative $\mathbf{S}^4$ criterion is inspired by the fourth-order weighting used in Kurtosis (Pearson, 1905), which places greater emphasis on large deviations. Since $\mathbf{S}^4$ amplifies large singular values relative to small singular values, we compute the ratio of the cumulative sum, $\sum_{i=1}^{k}\sigma_i^4$, to the total sum, $\sum_{i=1}^{r}\sigma_i^4$. For a given threshold τ, we select the smallest rank at which the cumulative ratio reaches

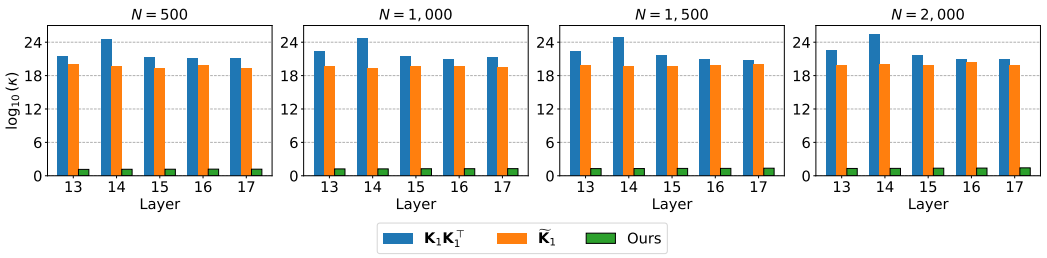

Figure 2: Comparison of the condition numbers of $\mathbf{K}_1\mathbf{K}_1^\top$, the centered key matrix $\widetilde{\mathbf{K}}_1$ before TSVD, and the centered key matrix after TSVD (Ours) on ZSRE using GPT-2 XL across the edited layers. N denotes the number of edits. Condition numbers are shown on a $\log_{10}$ scale.

or exceeds τ. In the 10,000-edit setting, the default threshold of $\tau = 0.9999$, i.e., 99.99%, yields a rank of 3,451. Additional sensitivity analysis and a comparison of the two criteria are provided in Appendices D and E, respectively.

Fig. 2 shows the improvement in the condition number across different editing scales. At the first edited layer (layer 13), MEMIT without $\mathbf{K}_0$ uses $\mathbf{K}_1\mathbf{K}_1^\top$, whose $\log_{10}(\kappa)$ ranges from 21.50 to 22.54 across 500 to 2,000 edits. The $\log_{10}(\kappa)$ of the centered key matrix $\widetilde{\mathbf{K}}_1$ before TSVD ranges from 19.65 to 20.00. After TSVD, $\log_{10}(\kappa)$ falls within the range of 1.15 to 1.31. Overall, the condition number of $\widetilde{\mathbf{K}}_1$ in the proposed method is lower than that of $\mathbf{K}_1\mathbf{K}_1^\top$ even before TSVD. Unlike unlearning approaches that aim to erase or suppress knowledge stored in model parameters, our method is applied only to newly collected key representations $\mathbf{K}_1$.

Table 2: Editing performance of the proposed method not utilizing $\mathbf{K}_0$ (Ours) and baseline methods utilizing $\mathbf{K}_0$ on ZSRE and MCF. N denotes the number of edits. Eff., Gen., Spec., Flu., and Cons. denote efficacy, generalization, specificity, fluency, and consistency, respectively. The best results are highlighted in bold.

Model	Method	$\mathbf{K}_0$	**ZSRE** ($N = 2,000$)			**MCF** ($N = 2,000$)					
			Eff.↑	Gen.↑	Spec.↑	Score↑	Eff.↑	Gen.↑	Spec.↑	Flu.↑	Cons.↑
Llama 3	MEMIT	✓	89.7$_{\pm 0.1}$	85.5$_{\pm 0.1}$	**32.6**$_{\pm 0.0}$	88.0$_{\pm 0.1}$	97.1$_{\pm 0.1}$	80.6$_{\pm 0.2}$	88.0$_{\pm 0.0}$	633.3$_{\pm 0.2}$	30.5$_{\pm 0.0}$
	PRUNE	✓	84.4$_{\pm 0.2}$	80.9$_{\pm 0.3}$	31.5$_{\pm 0.1}$	89.1$_{\pm 0.1}$	**99.2**$_{\pm 0.1}$	**94.7**$_{\pm 0.5}$	76.7$_{\pm 0.1}$	629.4$_{\pm 0.2}$	**35.3**$_{\pm 0.1}$
	RECT	✓	77.2$_{\pm 0.2}$	71.2$_{\pm 0.1}$	32.3$_{\pm 0.0}$	67.3$_{\pm 0.2}$	72.8$_{\pm 0.2}$	51.1$_{\pm 0.3}$	**88.7**$_{\pm 0.0}$	**634.6**$_{\pm 0.2}$	27.0$_{\pm 0.0}$
	AlphaEdit	✓	87.8$_{\pm 0.1}$	84.2$_{\pm 0.1}$	**32.6**$_{\pm 0.0}$	85.7$_{\pm 0.1}$	95.0$_{\pm 0.1}$	76.7$_{\pm 0.4}$	87.4$_{\pm 0.0}$	632.8$_{\pm 0.1}$	29.7$_{\pm 0.1}$
	MEMIT	×	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	49.7$_{\pm 0.6}$	49.9$_{\pm 1.2}$	49.6$_{\pm 1.2}$	49.5$_{\pm 0.5}$	424.1$_{\pm 68.2}$	2.3$_{\pm 1.2}$
	Ours	×	**90.7**$_{\pm 0.2}$	**86.8**$_{\pm 0.1}$	32.1$_{\pm 0.1}$	**91.7**$_{\pm 0.0}$	99.0$_{\pm 0.0}$	90.5$_{\pm 0.1}$	86.5$_{\pm 0.0}$	630.5$_{\pm 0.2}$	32.2$_{\pm 0.1}$
GPT-J	MEMIT	✓	98.4$_{\pm 0.0}$	94.4$_{\pm 0.1}$	27.7$_{\pm 0.1}$	**89.9**$_{\pm 0.1}$	**99.9**$_{\pm 0.0}$	92.8$_{\pm 0.2}$	79.5$_{\pm 0.1}$	621.2$_{\pm 0.2}$	40.8$_{\pm 0.1}$
	PRUNE	✓	74.6$_{\pm 0.8}$	70.0$_{\pm 0.8}$	25.8$_{\pm 0.1}$	84.1$_{\pm 0.1}$	98.5$_{\pm 0.1}$	**96.3**$_{\pm 0.2}$	66.0$_{\pm 0.2}$	594.5$_{\pm 0.8}$	39.8$_{\pm 0.1}$
	RECT	✓	84.4$_{\pm 0.1}$	75.1$_{\pm 0.2}$	27.9$_{\pm 0.1}$	83.7$_{\pm 0.1}$	96.7$_{\pm 0.1}$	75.7$_{\pm 0.3}$	**81.4**$_{\pm 0.1}$	**623.2**$_{\pm 0.2}$	37.6$_{\pm 0.0}$
	AlphaEdit	✓	**99.2**$_{\pm 0.1}$	**95.0**$_{\pm 0.1}$	27.3$_{\pm 0.1}$	89.8$_{\pm 0.1}$	99.8$_{\pm 0.0}$	94.0$_{\pm 0.1}$	78.5$_{\pm 0.1}$	620.0$_{\pm 0.2}$	**41.1**$_{\pm 0.1}$
	MEMIT	×	–	–	–	49.5$_{\pm 0.4}$	49.3$_{\pm 0.8}$	48.6$_{\pm 0.6}$	50.7$_{\pm 0.5}$	523.4$_{\pm 61.1}$	3.2$_{\pm 1.8}$
	Ours	×	96.5$_{\pm 0.1}$	92.8$_{\pm 0.2}$	**28.5**$_{\pm 0.1}$	89.4$_{\pm 0.0}$	99.3$_{\pm 0.0}$	93.8$_{\pm 0.1}$	77.8$_{\pm 0.1}$	618.7$_{\pm 0.3}$	39.8$_{\pm 0.1}$
GPT-2 XL	MEMIT	✓	76.6$_{\pm 0.1}$	70.0$_{\pm 0.1}$	**26.3**$_{\pm 0.0}$	81.3$_{\pm 0.0}$	92.4$_{\pm 0.1}$	79.5$_{\pm 0.1}$	74.1$_{\pm 0.0}$	625.7$_{\pm 0.2}$	38.4$_{\pm 0.1}$
	PRUNE	✓	64.1$_{\pm 0.2}$	61.1$_{\pm 0.2}$	23.3$_{\pm 0.1}$	80.7$_{\pm 0.0}$	96.0$_{\pm 0.1}$	91.6$_{\pm 0.1}$	63.1$_{\pm 0.1}$	604.5$_{\pm 0.5}$	38.3$_{\pm 0.1}$
	RECT	✓	62.6$_{\pm 0.1}$	57.5$_{\pm 0.1}$	**26.3**$_{\pm 0.0}$	73.1$_{\pm 0.1}$	80.9$_{\pm 0.1}$	64.4$_{\pm 0.2}$	**76.2**$_{\pm 0.0}$	**627.7**$_{\pm 0.2}$	36.1$_{\pm 0.1}$
	AlphaEdit	✓	**89.1**$_{\pm 0.2}$	**80.4**$_{\pm 0.2}$	24.9$_{\pm 0.1}$	**85.3**$_{\pm 0.1}$	**99.6**$_{\pm 0.0}$	**93.2**$_{\pm 0.2}$	69.4$_{\pm 0.1}$	617.1$_{\pm 0.2}$	**41.2**$_{\pm 0.1}$
	MEMIT	×	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	0.1$_{\pm 0.1}$	49.8$_{\pm 0.6}$	49.7$_{\pm 1.5}$	49.3$_{\pm 1.1}$	50.5$_{\pm 0.7}$	581.2$_{\pm 18.5}$	3.9$_{\pm 6.8}$
	Ours	×	85.8$_{\pm 0.1}$	78.5$_{\pm 0.2}$	24.0$_{\pm 0.1}$	84.1$_{\pm 0.1}$	97.5$_{\pm 0.1}$	89.9$_{\pm 0.2}$	69.9$_{\pm 0.1}$	620.1$_{\pm 0.6}$	**41.2**$_{\pm 0.1}$

Model	Method	$\mathbf{K}_0$	**ZSRE** ($N = 10,000$)			**MCF** ($N = 10,000$)					
			Eff.↑	Gen.↑	Spec.↑	Score↑	Eff.↑	Gen.↑	Spec.↑	Flu.↑	Cons.↑
GPT-2 XL	MEMIT	✓	62.8$_{\pm 0.7}$	58.1$_{\pm 0.7}$	**25.8**$_{\pm 0.1}$	71.5$_{\pm 0.1}$	79.5$_{\pm 0.1}$	66.6$_{\pm 0.2}$	69.7$_{\pm 0.0}$	624.1$_{\pm 0.2}$	36.0$_{\pm 0.0}$
	PRUNE	✓	42.9$_{\pm 1.1}$	40.7$_{\pm 1.2}$	19.0$_{\pm 0.2}$	71.4$_{\pm 0.1}$	84.1$_{\pm 0.2}$	**77.6**$_{\pm 0.1}$	58.0$_{\pm 0.1}$	603.1$_{\pm 0.3}$	34.8$_{\pm 0.1}$
	RECT	✓	49.5$_{\pm 0.5}$	45.4$_{\pm 0.4}$	**25.8**$_{\pm 0.2}$	60.5$_{\pm 0.1}$	62.8$_{\pm 0.1}$	49.4$_{\pm 0.2}$	**74.4**$_{\pm 0.0}$	**625.3**$_{\pm 0.1}$	33.9$_{\pm 0.0}$
	AlphaEdit	✓	62.6$_{\pm 1.2}$	55.7$_{\pm 1.2}$	21.2$_{\pm 0.3}$	**76.5**$_{\pm 0.1}$	**93.1**$_{\pm 0.1}$	77.1$_{\pm 0.2}$	64.5$_{\pm 0.1}$	601.7$_{\pm 0.5}$	**38.4**$_{\pm 0.1}$
	MEMIT	×	0.1$_{\pm 0.0}$	0.1$_{\pm 0.0}$	0.7$_{\pm 0.1}$	53.7$_{\pm 0.2}$	56.5$_{\pm 0.5}$	51.8$_{\pm 0.3}$	52.9$_{\pm 0.3}$	635.5$_{\pm 1.4}$	33.2$_{\pm 0.9}$
	Ours	×	**64.2**$_{\pm 1.0}$	**58.6**$_{\pm 0.8}$	22.3$_{\pm 0.3}$	75.2$_{\pm 0.1}$	88.7$_{\pm 0.1}$	75.9$_{\pm 0.2}$	64.9$_{\pm 0.1}$	616.1$_{\pm 0.7}$	**38.4**$_{\pm 0.1}$

4 EXPERIMENTS

We conduct experiments using GPT-2 XL (1.5B) (Radford et al., 2019), GPT-J (6B) (Wang & Komatsuzaki, 2021), and Llama 3 (8B) (Grattafiori et al., 2024) on a single NVIDIA RTX A6000 or RTX 3090 GPU. To assess the effectiveness and robustness of the proposed method, we evaluate it on diverse benchmarks, including ZSRE (Zero-Shot Relation Extraction) (Levy et al., 2017) for closed-domain QA, MCF (Multi-CounterFact) (Meng et al., 2023) for counterfactual editing, MQuAKE (Multi-hop Question Answering for Knowledge Editing) (Zhong et al., 2023) for multi-hop reasoning, and GLUE (General Language Understanding Evaluation) (Wang et al., 2019) for downstream tasks. We compared the proposed method, which performs factual association editing without pretrained knowledge representations $\mathbf{K}_0$, against state-of-the-art baselines, i.e., MEMIT (Meng et al., 2023), PRUNE (Ma et al., 2025), RECT (Gu et al., 2024), and AlphaEdit (Fang et al., 2025), which fully utilize $\mathbf{K}_0$ containing 44 million hidden vectors. Extended editing results for 100, 1,000, and 2,000 edits are provided in Appendix B.

4.1 EDITING PERFORMANCE

Tab. 2 presents the large-scale editing performance for 2,000 and 10,000 edits on ZSRE and MCF, measured by score, efficacy, generalization, specificity, fluency, and consistency. Score denotes the harmonic mean of efficacy, generalization, and specificity. Efficacy, generalization, and specificity measure whether the edited model correctly recalls the target knowledge, applies the edit to paraphrased queries, and preserves unrelated facts, respectively. Fluency measures the linguistic quality of model outputs, while consistency evaluates whether the edited knowledge is reflected coherently across related prompts.

As shown in Tab. 2, the proposed method achieves the highest efficacy and generalization on ZSRE using Llama 3, while exhibiting only a slight degradation in specificity, despite not utilizing pretrained knowledge representations. For GPT-2 XL, the proposed method attains the best efficacy and generalization at 10,000 edits. Our method similarly attains the competitive editing efficacy and generalization on MCF using Llama 3, where both our method and PRUNE, which aim to improve the condition number, achieve improved overall performance. Overall, the proposed method achieves editing performance comparable to or exceeding that of baseline methods that explicitly utilize pretrained knowledge representations $\mathbf{K}_0$. These results demonstrate the effectiveness of the proposed approach in performing factual association editing without dependence on pretrained knowledge representations even at large scales.

Consistent with prior work (Wang et al., 2024), our results highlight the difficulty of jointly maximizing efficacy, generalization, and specificity. In particular, the relative efficacy achieved by the proposed method compared with baseline approaches appears to be related to the norm of the weight perturbation $\mathbf{\Delta}$. A larger perturbation

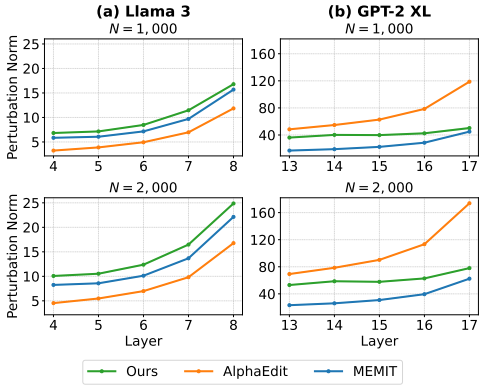

Figure 3: Layer-wise weight perturbation norms for Ours, AlphaEdit, and MEMIT on ZSRE using Llama 3 and GPT-2 XL.

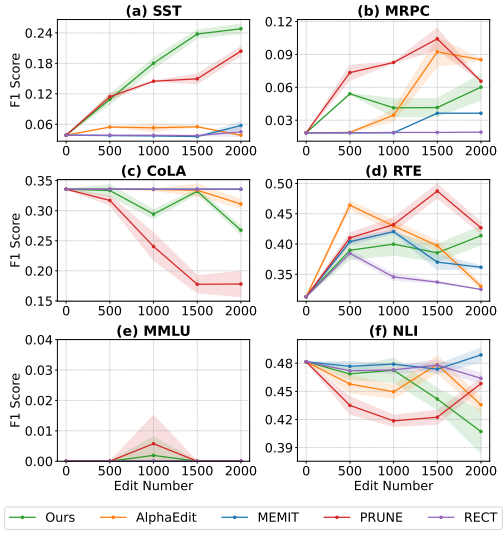

Figure 4: Downstream performance on six tasks across different numbers of edits using GPT-2 XL. Shaded regions indicate standard deviations across runs.

norm is associated with stronger modifications of the weight parameters. Conversely, a smaller perturbation norm results in more conservative updates. For instance, Fig. 3 shows that the norms of $\mathbf{\Delta}$ in GPT-2 XL induced by our method are larger than those of MEMIT, but remain smaller than those of AlphaEdit. On the other hand, the proposed method exhibits the largest norms in Llama 3. This aspect is aligned with the results in Tab. 2, indicating that larger weight perturbation norms generally improve efficacy, although at the expense of reduced specificity.

Fig. 4 presents a downstream evaluation on six tasks, i.e., SST, MRPC, CoLA, RTE, and NLI from the GLUE benchmark (Wang et al., 2019), and MMLU (Hendrycks et al., 2021). These tasks cover sentiment classification, paraphrase detection, grammatical acceptability, textual entailment, and multiple-choice reasoning. As shown in Fig. 4, the proposed method achieves overall competitive performance across these tasks. However, its performance on certain tasks remains lower than that of the best-performing baselines using $\mathbf{K}_0$. Since this evaluation assesses general language understanding, this degradation is likely attributable to side effects introduced during editing or to a loss of general capability, possibly stemming from the absence of pretrained knowledge representations.

Tab. 3 provides the evaluation results on the MQuAKE dataset, which assesses whether edited knowledge can be effectively propagated through multi-hop reasoning to produce correct answers. Although the proposed method achieves strong accuracy with up to 1,000 edits, its accuracy at 2,000 edits drops to nearly half of that observed at 1,000 edits, with no substantial difference between 2,000 and 3,000 edits. In conventional closed-form objectives, the $\mathbf{K}_0$ term penalizes the norm of $\mathbf{\Delta}\mathbf{K}_0$ to constrain changes in outputs associated with pretrained knowledge representations. At the last edited layer (layer 17), the relative perturbation norm, $\|\mathbf{\Delta}\|_F / \|\mathbf{W}\|_F$, increases from 1.98 to 3.67 for MEMIT and from 2.96 to 6.19 for our method as the number of edits increases from 1,000 to 2,000. The large relative perturbation norms suggest that weight perturbations whose norms exceed those of the model weights can degrade downstream multi-hop reasoning performance. These results demonstrate that further investigation is necessary to address this limitation.

Table 3: Multi-hop question answering accuracy of the proposed method not utilizing $\mathbf{K}_0$ (Ours) and baseline methods utilizing $\mathbf{K}_0$ on MQuAKE using GPT-2 XL. N denotes the number of edits. The best results are highlighted in bold.

Method	$\mathbf{K}_0$	**MQuAKE** (Accuracy ↑)		
		$N = 1,000$	$N = 2,000$	$N = 3,000$
MEMIT	✓	$7.0_{\pm 0.2}$	$\mathbf{6.1}_{\pm 0.4}$	$\mathbf{5.4}_{\pm 0.2}$
PRUNE	✓	$5.2_{\pm 0.2}$	$4.1_{\pm 0.1}$	$2.9_{\pm 0.1}$
RECT	✓	$6.8_{\pm 0.2}$	$5.5_{\pm 0.1}$	$4.9_{\pm 0.2}$
AlphaEdit	✓	$0.4_{\pm 0.1}$	$0.5_{\pm 0.2}$	$0.3_{\pm 0.1}$
MEMIT	×	$0.1_{\pm 0.1}$	$0.3_{\pm 0.1}$	$0.2_{\pm 0.1}$
Ours	×	$\mathbf{7.5}_{\pm 0.5}$	$3.8_{\pm 0.4}$	$3.6_{\pm 0.3}$

4.2 Efficiency Analysis

Tab. 4 compares the efficiency of the proposed method with baseline approaches that depend on pretrained knowledge representations $\mathbf{K}_0$. One-time precomputation requires approximately 17 GB of a Wikipedia corpus and takes 5-20 hours per model on a single NVIDIA RTX A6000 GPU. We measure the amortized runtime over ten runs. Each run uses the MCF dataset with a batch size of 100. For baseline approaches, the precomputation time is amortized over ten runs. The proposed method achieves the lowest amortized runtime across all three models, with speedups of 11.1-20.7× over MEMIT and 14.8-30.9× over AlphaEdit. Reusing pretrained knowledge representations requires additional persistent storage of 782 MiB for GPT-2 XL, 6.00 GiB for GPT-J, and 3.83 GiB

Table 4: Efficiency analysis of the proposed method not utilizing $\mathbf{K}_0$ (Ours) and baseline approaches utilizing $\mathbf{K}_0$. Precomp. Dataset denotes the size of a Wikipedia corpus used to precompute $\mathbf{C}_0 = \mathbf{K}_0\mathbf{K}_0^\top$. Precomp. Time denotes the approximate precomputation time. Amortized Runtime is measured for 100 edits on MCF, with the precomputation time of baseline approaches amortized over ten runs. Storage Cost denotes the additional persistent storage required to store $\mathbf{C}_0$. The lowest values are highlighted in bold.

Method	Precomp. Dataset (GiB) ↓	Precomp. Time (h) ↓			Amortized Runtime (min) ↓			Storage Cost (GiB) ↓		
		GPT-2	GPT-J	Llama 3	GPT-2	GPT-J	Llama 3	GPT-2	GPT-J	Llama 3
MEMIT	17	5	20	20	33.8	126.5	141.3	0.76	6.00	3.83
AlphaEdit	17	5	20	20	34.3	188.2	188.3	0.76	6.00	3.83
Ours	**0**	**0**	**0**	**0**	**2.3**	**6.1**	**12.7**	**0**	**0**	**0**

for Llama 3. The proposed method eliminates the need for the corpus, one-time precomputation, and additional persistent storage for pretrained knowledge representations. These results demonstrate that removing pretrained knowledge representations improves the efficiency and scalability of factual association editing.

5 Related Work

Closed-form editing analytically computes the weight perturbation to modify specific factual associations in a language model. ROME (Meng et al., 2022) is the first work to propose the locate-then-edit framework for knowledge editing, identifying critical layers for modification via causal tracing. MEMIT (Meng et al., 2023) extends this framework to enable large-scale memory editing, demonstrating that multiple factual associations can be edited simultaneously. PMET (Li et al., 2024) jointly optimizes the hidden states of Transformer Component while updating only FFN weights, suggesting that effective editing can be achieved without updating multi-head self-attention weights.

Subsequent methods improve editing performance through weight rectification (Gu et al., 2024), perturbation restraint (Ma et al., 2025), null-space projection (Fang et al., 2025), or mitigation of residual distribution errors (Li et al., 2025). FastMEMIT (Gupta et al., 2025) substantially reduces the number of hidden vectors to be precomputed but still relies on precomputed representations. UltraEdit (Gu et al., 2026) employs only a hidden state and its gradient to compute parameter shifts. Unlike UltraEdit, we adopt a linear associative memory approach in line with prior work (Meng et al., 2022; 2023; Fang et al., 2025). Furthermore, we do not maintain feature statistics across turns, relying solely on newly introduced knowledge.

Alternative approaches have also been explored; MEND (Mitchell et al., 2022a) employs auxiliary editing networks to make fast edits. SERAC (Mitchell et al., 2022b) stores edits in an explicit memory, uses a classifier to determine whether a relevant edit exists, and passes the edit to the counterfactual model. GRACE (Hartvigsen et al., 2023) creates a discrete codebook of edits, MELO (Yu et al., 2024) dynamically activates LoRA blocks based on the index in a vector database, and WISE (Wang et al., 2024) combines a side memory for the edited knowledge with a routing mechanism. These methods rely on auxiliary editing networks, routers, or side memories.

In existing approaches to closed-form factual association editing (Meng et al., 2023; Fang et al., 2025), pretrained knowledge representations $\mathbf{K}_0$ are typically constructed from a subset of the target dataset, e.g., by randomly sampling 100,000 instances from the snapshot of Wikipedia (Vrandecić & Krötzsch, 2014). Hence, even editing a relatively small number of factual associations (e.g., ten facts) incurs substantially high memory and computational costs, as $\mathbf{K}_0\mathbf{K}_0^\top$ must be retained and repeatedly reused for both current and future editing. In contrast, the proposed method overcomes this limitation by completely removing the dependence on $\mathbf{K}_0$, resulting in more practical, scalable, and efficient factual association editing. Specifically, the proposed method reduces the condition number by within-representation centering and TSVD, effectively addressing more realistic and challenging scenarios induced by the absence of pretrained representations.

6 Conclusion

We propose a method for closed-form factual association editing that eliminates the need for pretrained knowledge representations, addressing a fundamental limitation of existing approaches. The proposed method improves editing stability through within-representation centering and TSVD by handling ill-conditioning and the effects of singular values. Unlike prior methods relying on proxy datasets to approximate pretrained knowledge, our approach achieves scalable, memory-efficient editing without such dependencies. Experimental results demonstrate that the proposed method attains comparable editing accuracy to state-of-the-art approaches using pretrained knowledge representations at large editing scales, demonstrating the feasibility of performing reliable and effective factual association editing without them.

AI use statement

In this work, we used generative AI tools to assist with translation. We have not used generative AI tools to propose hypotheses, design research methodology or interpret results. Generating syn-

thetic datasets is not applicable to this work. Additionally, we used generative AI tools for creating scientific figures, editing software code, summarizing existing literature and formatting references. We have reviewed all AI-assisted work. We checked LLM-generated code for consistency with the formula. Scientific figures were created according to detailed instructions provided by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Although the proposed approach enables editing factual associations without relying on pretrained knowledge representations, it may also introduce potential ethical risks. Specifically, malicious actors could exploit this approach to inject misinformation or harmful content into language models. Therefore, responsible deployment and strict validation mechanisms are necessary to mitigate these risks.

REPRODUCIBILITY STATEMENT

The experimental setup is described in Section 4. We use the same hyperparameter settings as AlphaEdit (Fang et al., 2025), except for the hyperparameter newly introduced in Section 3.4. Limitations are discussed in Appendix A. The source code is available in an anonymous repository[1].

REFERENCES

Daniel Biś, Maksim Podkorytov, and Xiuwen Liu. Too much in common: Shifting of embeddings in transformer language models and its implications. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5117–5130, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.403. URL https://aclanthology.org/2021.naacl-main.403/.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020. URL https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45, 2024. doi: 10.1145/3641289. URL https://dl.acm.org/doi/10.1145/3641289.

Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 55–65, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1006. URL https://aclanthology.org/D19-1006/.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. AlphaEdit: Null-space constrained knowledge editing for language models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HvSytvg3Jh.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. doi: 10.48550/arXiv.2407.21783. URL https://arxiv.org/abs/2407.21783.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. Model editing harms general abilities of large language models: Regularization to the rescue. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.934. URL https://aclanthology.org/2024.emnlp-main.934/.

Xiaojie Gu, Ziying Huang, Jia-Chen Gu, and Kai Zhang. UltraEdit: Training-, subject-, and memory-free lifelong editing in language models. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=GoJLp3BlRV.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10746–10761. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.568. URL https://aclanthology.org/2025.findings-emnlp.568/.

Akshat Gupta, Maochuan Lu, Thomas Hartvigsen, and Gopala Anumanchipalli. Efficient knowledge editing via minimal precomputation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 829–840, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-short.65. URL https://aclanthology.org/2025.acl-short.65/.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. REALM: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 3929–3938. PMLR, 2020. URL https://proceedings.mlr.press/v119/guu20a.html.

Per Christian Hansen. The truncated SVD as a method for regularization. *BIT Numerical Mathematics*, 27:534–553, 1987. doi: 10.1007/BF01937276. URL https://doi.org/10.1007/BF01937276.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with GRACE: Lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023. doi: 10.52202/075280-2079. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/95b6e2ff961580e03c0a662a63a71812-Abstract-Conference.html.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35. Curran Associates, Inc., 2022. doi: 10.52202/068431-1613. URL https://proceedings.neurips.cc/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pp. 333–342, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1034. URL https://aclanthology.org/K17-1034/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020. URL https://papers.nips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

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, pp. 18564–18572. AAAI Press, 2024. doi: 10.1609/aaai.v38i17.29818. URL https://ojs.aaai.org/index.php/AAAI/article/view/29818.

Xiaopeng Li, Shangwen Wang, Shasha Li, Shezheng Song, Bin Ji, Jun Ma, and Jie Yu. Rethinking residual distribution in locate-then-edit model editing. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=P9gY05BDkW.

Jun-Yu Ma, Hong Wang, Hao-Xiang Xu, Zhen-Hua Ling, and Jia-Chen Gu. Perturbation-restrained sequential model editing. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/2c15b0221da28bc6f4373a7e78b896dd-Abstract-Conference.html.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372. Curran Associates, Inc., 2022. doi: 10.52202/068431-1262. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=0DcZxeWfOPt.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15817–15831. PMLR, 2022b. URL https://proceedings.mlr.press/v162/mitchell22a.html.

Eliakim H Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the american mathematical society*, 26:294–295, 1920.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkuGJ3kCb.

Karl Pearson. "das fehlergesetz und seine verallgemeiner-ungen durch fechner und pearson." a rejoinder. *Biometrika*, 4(1-2):169–212, 1905. doi: 10.1093/biomet/4.1-2.169. URL https://academic.oup.com/biomet/article/4/1-2/169/186085.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *European Signal Processing Conference*, pp. 606–610, Poznan, Poland, 2007. URL https://www.eurasip.org/Proceedings/Eusipco/Eusipco2007/Papers/a5p-h05.pdf.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. Editable neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJedXaEtvS.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://papers.nips.cc/paper/7181-attention-is-all-you-need.

Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014. doi: 10.1145/2629489. URL https://doi.org/10.1145/2629489.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 billion parameter autoregressive language model, May 2021. URL https://github.com/kingoflolz/mesh-transformer-jax.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. WISE: Rethinking the knowledge memory for lifelong model editing of large language models. In *Advances in Neural Information Processing Systems*, volume 37. Curran Associates, Inc., 2024. doi: 10.52202/079017-1703. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/60960ad78868fce5c165295fbd895060-Abstract-Conference.html.

Zihan Wang, Chengyu Dong, and Jingbo Shang. "average" approximates "first principal component"? an empirical analysis on representations from neural language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5594–5603, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.453. URL https://aclanthology.org/2021.emnlp-main.453/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. MELO: Enhancing model editing with neuron-indexed dynamic LoRA. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19449–19457. AAAI Press, 2024. doi: 10.1609/aaai.v38i17.29916. URL https://ojs.aaai.org/index.php/AAAI/article/view/29916.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15686–15702. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.971. URL https://aclanthology.org/2023.emnlp-main.971/.

A Limitations and Discussions

Although the proposed method investigates the feasibility of closed-form factual association editing without using pretrained knowledge representations, this research direction still presents limitations that require further study.

First, specificity, defined as the ability to preserve pretrained knowledge, tends to deteriorate as the number of edits increases, due to the absence of explicit mechanisms for retaining existing knowledge, which is an expected behavior. To mitigate this issue, one potential direction may include incorporating a limited amount of pretrained knowledge, rather than a large volume, e.g., in the form of representative representations, to facilitate the recovery of lost knowledge. This approach would enable the model to better retain pretrained knowledge over a large number of edits while preserving scalability, adaptability, and efficiency in the editing process.

Second, this work primarily focuses on one-shot batch editing, i.e., updating a set of multiple factual associations simultaneously. As a natural extension, it can be generalized to sequential editing. However, because sequential editing involves multiple successive model updates, it is inherently more susceptible to degradation of pretrained knowledge due to repeated modifications of the model parameters.

Third, the proposed method can be further improved along several dimensions. For instance, rather than updating the entire weight $\mathbf{W}$ of the target layers, selectively updating a subset of weights may better preserve pretrained knowledge. In addition, it can be extended to general knowledge editing settings beyond factual associations, enabling the model to accommodate diverse forms of knowledge updates without explicitly retaining prior knowledge.

Lastly, reliable model performance on downstream tasks after factual association editing needs to be preserved to maintain general language capabilities, which has been shown to be a limitation in our experiments.

B Additional Experiments

We conduct experiments on the ZSRE dataset with batch sizes of 100, 1,000, and 2,000 edits. We also compare the proposed method with FastMEMIT (Gupta et al., 2025), a minimal-precomputation approach. As shown in Tab. 5, the proposed method achieves performance comparable to Fast-

Table 5: Batch editing performance of the proposed method not utilizing $\mathbf{K}_0$ (Ours) and baseline approaches utilizing $\mathbf{K}_0$ on ZSRE. N denotes the number of edits. Eff., Gen., and Spec. denote efficacy, generalization, and specificity, respectively. The best results are highlighted in bold.

Model	Method	$\mathbf{K}_0$	ZSRE								
			$N = 100$			$N = 1,000$			$N = 2,000$		
			Eff.↑	Gen.↑	Spec.↑	Eff.↑	Gen.↑	Spec.↑	Eff.↑	Gen.↑	Spec.↑
Llama 3	MEMIT	✓	$83.5_{\pm 0.0}$	$80.9_{\pm 0.2}$	$\mathbf{33.0}_{\pm 0.0}$	$90.4_{\pm 0.1}$	$86.8_{\pm 0.1}$	$\mathbf{33.0}_{\pm 0.1}$	$89.7_{\pm 0.1}$	$85.5_{\pm 0.1}$	$\mathbf{32.6}_{\pm 0.0}$
	PRUNE	✓	$91.8_{\pm 0.3}$	$\mathbf{89.4}_{\pm 0.1}$	$32.4_{\pm 0.1}$	$89.6_{\pm 0.1}$	$87.4_{\pm 0.2}$	$32.6_{\pm 0.1}$	$84.4_{\pm 0.2}$	$80.9_{\pm 0.3}$	$31.5_{\pm 0.1}$
	RECT	✓	$68.8_{\pm 0.0}$	$63.9_{\pm 0.2}$	$32.8_{\pm 0.0}$	$79.8_{\pm 0.2}$	$73.9_{\pm 0.2}$	$\mathbf{33.0}_{\pm 0.0}$	$77.2_{\pm 0.2}$	$71.2_{\pm 0.1}$	$32.3_{\pm 0.0}$
	AlphaEdit	✓	$85.3_{\pm 0.0}$	$83.0_{\pm 0.2}$	$32.7_{\pm 0.1}$	$89.8_{\pm 0.1}$	$86.4_{\pm 0.1}$	$32.8_{\pm 0.1}$	$87.8_{\pm 0.1}$	$84.2_{\pm 0.1}$	$\mathbf{32.6}_{\pm 0.0}$
	FastMEMIT	✓	$86.5_{\pm 0.5}$	$84.2_{\pm 0.2}$	$32.9_{\pm 0.2}$	$\mathbf{91.5}_{\pm 0.1}$	$87.9_{\pm 0.1}$	$\mathbf{33.0}_{\pm 0.0}$	$90.5_{\pm 0.1}$	$86.5_{\pm 0.1}$	$\mathbf{32.6}_{\pm 0.0}$
	Ours	×	$\mathbf{92.5}_{\pm 0.4}$	$88.5_{\pm 0.6}$	$32.2_{\pm 0.0}$	$91.3_{\pm 0.1}$	$\mathbf{88.9}_{\pm 0.1}$	$32.8_{\pm 0.1}$	$\mathbf{90.7}_{\pm 0.2}$	$\mathbf{86.8}_{\pm 0.1}$	$32.1_{\pm 0.1}$
GPT-J	MEMIT	✓	$99.3_{\pm 0.0}$	$95.3_{\pm 0.1}$	$28.3_{\pm 0.1}$	$99.0_{\pm 0.0}$	$95.4_{\pm 0.1}$	$28.3_{\pm 0.1}$	$98.4_{\pm 0.0}$	$94.4_{\pm 0.1}$	$27.7_{\pm 0.1}$
	PRUNE	✓	$97.9_{\pm 0.0}$	$96.6_{\pm 0.0}$	$\mathbf{28.6}_{\pm 0.1}$	$91.7_{\pm 0.3}$	$88.4_{\pm 0.4}$	$28.1_{\pm 0.2}$	$74.6_{\pm 0.8}$	$70.0_{\pm 0.8}$	$25.8_{\pm 0.1}$
	RECT	✓	$92.8_{\pm 0.1}$	$83.8_{\pm 0.4}$	$28.4_{\pm 0.0}$	$87.1_{\pm 0.3}$	$78.3_{\pm 0.3}$	$28.0_{\pm 0.1}$	$84.4_{\pm 0.1}$	$75.1_{\pm 0.2}$	$27.9_{\pm 0.1}$
	AlphaEdit	✓	$99.4_{\pm 0.1}$	$97.0_{\pm 0.0}$	$28.5_{\pm 0.1}$	$\mathbf{99.5}_{\pm 0.0}$	$\mathbf{96.1}_{\pm 0.1}$	$27.4_{\pm 0.0}$	$\mathbf{99.2}_{\pm 0.1}$	$\mathbf{95.0}_{\pm 0.1}$	$27.3_{\pm 0.1}$
	FastMEMIT	✓	$86.9_{\pm 0.3}$	$82.0_{\pm 0.0}$	$28.4_{\pm 0.0}$	$89.8_{\pm 0.1}$	$84.2_{\pm 0.3}$	$28.5_{\pm 0.0}$	$89.4_{\pm 0.1}$	$83.7_{\pm 0.1}$	$\mathbf{29.0}_{\pm 0.0}$
	Ours	×	$\mathbf{99.5}_{\pm 0.2}$	$\mathbf{97.8}_{\pm 0.1}$	$28.2_{\pm 0.1}$	$98.2_{\pm 0.0}$	$95.1_{\pm 0.1}$	$\mathbf{28.7}_{\pm 0.1}$	$96.5_{\pm 0.1}$	$92.8_{\pm 0.2}$	$28.5_{\pm 0.1}$
GPT-2 XL	MEMIT	✓	$83.8_{\pm 0.1}$	$77.4_{\pm 0.3}$	$26.1_{\pm 0.0}$	$79.6_{\pm 0.1}$	$74.0_{\pm 0.2}$	$\mathbf{26.2}_{\pm 0.0}$	$76.6_{\pm 0.1}$	$70.0_{\pm 0.1}$	$\mathbf{26.3}_{\pm 0.0}$
	PRUNE	✓	$87.8_{\pm 0.4}$	$79.8_{\pm 0.3}$	$\mathbf{27.3}_{\pm 0.1}$	$72.9_{\pm 0.2}$	$70.4_{\pm 0.2}$	$25.2_{\pm 0.1}$	$64.1_{\pm 0.2}$	$61.1_{\pm 0.2}$	$23.3_{\pm 0.1}$
	RECT	✓	$72.4_{\pm 0.2}$	$64.6_{\pm 0.5}$	$25.1_{\pm 0.0}$	$65.0_{\pm 0.2}$	$60.5_{\pm 0.1}$	$25.6_{\pm 0.0}$	$62.6_{\pm 0.1}$	$57.5_{\pm 0.1}$	$\mathbf{26.3}_{\pm 0.0}$
	AlphaEdit	✓	$\mathbf{95.1}_{\pm 0.2}$	$\mathbf{89.3}_{\pm 0.4}$	$25.6_{\pm 0.4}$	$\mathbf{93.8}_{\pm 0.1}$	$\mathbf{86.7}_{\pm 0.1}$	$25.8_{\pm 0.1}$	$\mathbf{89.1}_{\pm 0.2}$	$\mathbf{80.4}_{\pm 0.2}$	$24.9_{\pm 0.1}$
	FastMEMIT	✓	$85.4_{\pm 0.1}$	$79.7_{\pm 0.5}$	$25.8_{\pm 0.1}$	$82.8_{\pm 0.1}$	$76.7_{\pm 0.1}$	$\mathbf{26.2}_{\pm 0.0}$	$78.4_{\pm 0.1}$	$71.8_{\pm 0.1}$	$\mathbf{26.3}_{\pm 0.1}$
	Ours	×	$92.4_{\pm 0.3}$	$88.1_{\pm 0.3}$	$27.1_{\pm 0.3}$	$89.0_{\pm 0.2}$	$82.1_{\pm 0.3}$	$24.1_{\pm 0.2}$	$85.8_{\pm 0.1}$	$78.5_{\pm 0.2}$	$24.0_{\pm 0.1}$

Table 6: Batch editing performance of the proposed method and UltraEdit on ZSRE using Mistral. N denotes the number of edits. Eff., Gen., and Spec. denote efficacy, generalization, and specificity, respectively. The best results are highlighted in bold.

Model	Method	ZSRE					
		$N = 100$			$N = 1,000$		
		Eff.↑	Gen.↑	Spec.↑	Eff.↑	Gen.↑	Spec.↑
Mistral	UltraEdit	56.5	52.8	49.0	59.0	56.2	47.7
	Ours	**97.9**	**88.8**	**50.7**	**97.6**	**89.1**	**49.0**

Table 7: Sequential editing performance of the proposed method and UltraEdit on ZSRE using Mistral. N denotes the number of edits. Eff., Gen., and Spec. denote efficacy, generalization, and specificity, respectively. Ours* denotes the variant of the proposed method used for sequential editing. The best results are highlighted in bold.

Model	Method	ZSRE					
		$N = 100$			$N = 1,000$		
		Eff.↑	Gen.↑	Spec.↑	Eff.↑	Gen.↑	Spec.↑
Mistral	UltraEdit	55.4	53.6	48.6	**68.8**	**62.5**	**47.8**
	Ours*	**96.5**	**90.4**	**50.5**	24.5	23.1	23.5

MEMIT despite completely eliminating pretrained knowledge representations $\mathbf{K}_0$. For GPT-2 XL, it consistently achieves the second-best efficacy and generalization across all three edit sizes. These results demonstrate that the proposed method remains effective across models of different sizes.

We evaluate the proposed method using Mistral (7B) with UltraEdit (Gu et al., 2026) in both batch and sequential editing settings. For sequential editing, we adapt the proposed method to remove the first principal component of the representations for mitigating forgetting of pretrained knowledge. We conduct sequential editing with batch sizes of 10 and 100 for $N = 100$ and $N = 1,000$, respectively. As shown in Tabs. 6 and 7, while the proposed method outperforms UltraEdit in batch editing, it is less effective for large-scale sequential editing. It is necessary to address this limitation in future research.

C Ablation Study

We conducted an ablation study to evaluate the contribution of each component in the proposed method for factual association editing without relying on pretrained knowledge representations, as summarized in Tab. 8. Starting from vanilla MEMIT (w/ $\mathbf{K}_0$) (Meng et al., 2023), we first remove $\mathbf{K}_0$ (w/o $\mathbf{K}_0$), then use the Moore–Penrose pseudoinverse $\mathbf{K}_1^\dagger$. Finally, we employ TSVD to obtain Ours (w/o centering), followed by applying within-representation centering (Ours (all)).

Removing $\mathbf{K}_0$ from MEMIT causes editing performance to collapse at N=100 and N=1,000. The proposed method improves editing performance for $N = 100$ and $N = 1,000$ over MEMIT with $\mathbf{K}_1^\dagger$. Furthermore, the performance of MEMIT with $\mathbf{K}_1^\dagger$ approaches zero for $N = 10,000$, yielding an efficacy of 0.1, generalization of 0.2, and specificity of 0.7. As shown in Fig. 5, the full proposed method yields a lower condition number than Ours without centering, indicating that within-representation centering further reduces the condition number. These results demonstrate the effectiveness of the proposed method for stable editing in the absence of $\mathbf{K}_0$, particularly as the number of edits increases.

Table 8: Ablation study of the proposed method (Ours) on ZSRE using GPT-2 XL. N denotes the number of edits. Eff., Gen., and Spec. denote efficacy, generalization, and specificity, respectively.

Method	ZSRE								
	$N = 1$			$N = 100$			$N = 1,000$		
	Eff.↑	Gen.↑	Spec.↑	Eff.↑	Gen.↑	Spec.↑	Eff.↑	Gen.↑	Spec.↑
MEMIT (w/ $\mathbf{K}_0$)	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$66.7_{\pm 0.0}$	$83.8_{\pm 0.1}$	$77.4_{\pm 0.3}$	$26.1_{\pm 0.0}$	$79.6_{\pm 0.1}$	$74.0_{\pm 0.2}$	$26.2_{\pm 0.0}$
MEMIT (w/o $\mathbf{K}_0$)	–	–	–	$0.1_{\pm 0.3}$	$0.2_{\pm 0.6}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
MEMIT (w/ $\mathbf{K}_1^\dagger$)	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$66.7_{\pm 0.0}$	$88.6_{\pm 0.5}$	$80.9_{\pm 0.3}$	$26.8_{\pm 0.3}$	$73.3_{\pm 0.4}$	$65.2_{\pm 0.3}$	$21.8_{\pm 0.2}$
Ours (w/o centering)	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$66.7_{\pm 0.0}$	$90.6_{\pm 0.6}$	$82.2_{\pm 1.0}$	$26.8_{\pm 0.4}$	$81.5_{\pm 0.2}$	$75.3_{\pm 0.2}$	$24.5_{\pm 0.1}$
Ours (all)	$100.0_{\pm 0.0}$	$100.0_{\pm 0.0}$	$66.7_{\pm 0.0}$	$92.4_{\pm 0.3}$	$88.1_{\pm 0.3}$	$27.1_{\pm 0.3}$	$89.0_{\pm 0.2}$	$82.1_{\pm 0.3}$	$24.1_{\pm 0.2}$

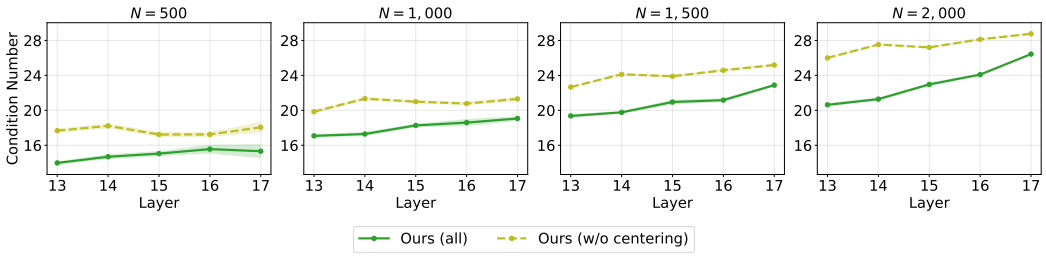

Figure 5: Comparison of the condition numbers of Ours (w/o centering) and Ours (all) on ZSRE using GPT-2 XL. Shaded regions indicate the standard deviation across runs.

D SENSITIVITY ANALYSIS

We also analyzed the sensitivity of the cumulative $\mathbf{S}^4$ criterion to the threshold hyperparameter. Tab. 9 reports the retained ranks for 100, 1,000, 2,000, and 10,000 edits. We conduct experiments on the ZSRE dataset across different thresholds for 100, 1,000, and 2,000 edits. As shown in Tab. 10, editing performance remains stable for thresholds ranging from 0.9995 to 0.99995, whereas a lower threshold of 0.995 generally leads to performance degradation. These results show that overly aggressive truncation can adversely affect editing performance. Overall, the method is relatively robust to the threshold choice within this range, with the default threshold of 0.9999 consistently providing strong performance.

Table 9: Ranks retained by the cumulative $\mathbf{S}^4$ criterion at the first edited layer (layer 13) on ZSRE using GPT-2 XL across different thresholds. N denotes the number of edits. In the main experiments, the default threshold is set to 0.9999.

Edit Number	Truncation Threshold				
	0.995	0.999	0.9995	**0.9999**	0.99995
100	45	50	52	63	68
1,000	381	477	502	548	578
2,000	679	911	981	1,089	1,135
10,000	1,560	2,397	2,735	3,451	3,730

Table 10: Editing performance using the cumulative $\mathbf{S}^4$ criterion on ZSRE across different thresholds. N denotes the number of edits. Eff., Gen., and Spec. denote efficacy, generalization, and specificity, respectively. In the main experiments, the default threshold is set to 0.9999. The best results are highlighted in bold.

Model	Threshold	ZSRE								
		$N = 100$			$N = 1,000$			$N = 2,000$		
		Eff.↑	Gen.↑	Spec.↑	Eff.↑	Gen.↑	Spec.↑	Eff.↑	Gen.↑	Spec.↑
Llama 3	0.995	83.0$_{\pm 0.4}$	78.9$_{\pm 0.3}$	32.1$_{\pm 0.0}$	82.6$_{\pm 0.1}$	80.8$_{\pm 0.1}$	**33.0**$_{\pm 0.0}$	76.7$_{\pm 0.2}$	74.6$_{\pm 0.2}$	**32.6**$_{\pm 0.0}$
	0.999	87.2$_{\pm 0.3}$	83.1$_{\pm 0.0}$	32.0$_{\pm 0.1}$	89.8$_{\pm 0.1}$	87.0$_{\pm 0.0}$	32.8$_{\pm 0.1}$	86.6$_{\pm 0.1}$	83.4$_{\pm 0.1}$	32.3$_{\pm 0.1}$
	0.9995	88.2$_{\pm 0.3}$	83.8$_{\pm 0.3}$	31.9$_{\pm 0.0}$	90.6$_{\pm 0.3}$	87.9$_{\pm 0.1}$	32.7$_{\pm 0.0}$	88.6$_{\pm 0.1}$	85.2$_{\pm 0.2}$	32.2$_{\pm 0.1}$
	0.9999	**92.5**$_{\pm 0.6}$	**88.4**$_{\pm 0.8}$	**32.2**$_{\pm 0.0}$	**91.3**$_{\pm 0.2}$	**89.0**$_{\pm 0.0}$	32.9$_{\pm 0.1}$	**90.9**$_{\pm 0.2}$	86.8$_{\pm 0.1}$	32.1$_{\pm 0.0}$
	0.99995	87.6$_{\pm 0.0}$	85.3$_{\pm 0.9}$	32.1$_{\pm 0.1}$	90.9$_{\pm 0.1}$	88.5$_{\pm 0.1}$	32.8$_{\pm 0.0}$	90.5$_{\pm 0.1}$	**87.1**$_{\pm 0.1}$	32.4$_{\pm 0.0}$
GPT-J	0.995	96.4$_{\pm 0.0}$	95.8$_{\pm 0.2}$	27.7$_{\pm 0.0}$	88.5$_{\pm 0.1}$	85.1$_{\pm 0.0}$	**29.1**$_{\pm 0.0}$	82.2$_{\pm 0.2}$	79.0$_{\pm 0.1}$	**29.6**$_{\pm 0.1}$
	0.999	98.1$_{\pm 0.0}$	97.7$_{\pm 0.0}$	28.2$_{\pm 0.0}$	95.7$_{\pm 0.1}$	92.3$_{\pm 0.1}$	28.8$_{\pm 0.1}$	92.2$_{\pm 0.2}$	88.2$_{\pm 0.1}$	29.1$_{\pm 0.1}$
	0.9995	99.2$_{\pm 0.0}$	**97.9**$_{\pm 0.0}$	27.6$_{\pm 0.1}$	96.5$_{\pm 0.0}$	93.4$_{\pm 0.1}$	28.7$_{\pm 0.0}$	94.2$_{\pm 0.1}$	90.0$_{\pm 0.1}$	28.5$_{\pm 0.0}$
	0.9999	**99.4**$_{\pm 0.2}$	97.6$_{\pm 0.2}$	**28.3**$_{\pm 0.2}$	**98.2**$_{\pm 0.0}$	**95.2**$_{\pm 0.1}$	28.7$_{\pm 0.1}$	96.4$_{\pm 0.1}$	**92.8**$_{\pm 0.3}$	28.5$_{\pm 0.0}$
	0.99995	98.7$_{\pm 0.0}$	97.5$_{\pm 0.1}$	27.9$_{\pm 0.0}$	98.0$_{\pm 0.2}$	94.7$_{\pm 0.2}$	28.5$_{\pm 0.1}$	**96.5**$_{\pm 0.0}$	**92.8**$_{\pm 0.1}$	28.2$_{\pm 0.1}$
GPT-2 XL	0.995	88.2$_{\pm 0.2}$	85.0$_{\pm 0.3}$	26.5$_{\pm 0.2}$	81.5$_{\pm 0.2}$	76.7$_{\pm 0.2}$	**25.8**$_{\pm 0.0}$	74.5$_{\pm 0.2}$	69.9$_{\pm 0.1}$	**26.2**$_{\pm 0.0}$
	0.999	90.6$_{\pm 0.0}$	87.5$_{\pm 0.5}$	26.8$_{\pm 0.2}$	86.6$_{\pm 0.2}$	81.5$_{\pm 0.2}$	25.0$_{\pm 0.1}$	82.1$_{\pm 0.2}$	75.5$_{\pm 0.2}$	24.8$_{\pm 0.1}$
	0.9995	91.1$_{\pm 0.2}$	87.1$_{\pm 0.4}$	27.0$_{\pm 0.3}$	87.8$_{\pm 0.0}$	82.1$_{\pm 0.2}$	24.9$_{\pm 0.1}$	83.9$_{\pm 0.1}$	76.8$_{\pm 0.1}$	24.6$_{\pm 0.1}$
	0.9999	92.4$_{\pm 0.3}$	88.0$_{\pm 0.2}$	**27.2**$_{\pm 0.1}$	89.0$_{\pm 0.4}$	81.8$_{\pm 0.2}$	24.1$_{\pm 0.3}$	85.8$_{\pm 0.1}$	78.5$_{\pm 0.2}$	24.1$_{\pm 0.1}$
	0.99995	**92.5**$_{\pm 0.3}$	**89.3**$_{\pm 0.6}$	26.6$_{\pm 0.6}$	**89.2**$_{\pm 0.3}$	**82.8**$_{\pm 0.0}$	24.7$_{\pm 0.2}$	**86.4**$_{\pm 0.3}$	**79.0**$_{\pm 0.2}$	23.7$_{\pm 0.1}$

E Comparison of Truncation Criteria

We introduce the spectral entropy criterion and the cumulative $\mathbf{S}^4$ criterion to determine the retained rank. To evaluate the effectiveness of two criteria, we conduct experiments on the ZSRE and MQuAKE datasets across different edit sizes. Fig. 6 shows that the spectral entropy criterion retains more small singular values than the cumulative $\mathbf{S}^4$ criterion. As shown in Tab. 11, the cumulative $\mathbf{S}^4$ criterion achieves comparable or better results than the spectral entropy criterion across different edit sizes. Both criteria effectively determine the retained rank for 100 and 1,000 edits. However, the cumulative $\mathbf{S}^4$ criterion performs better for 10,000 edits. These results suggest that retaining more small singular values is largely harmless at smaller scales but may amplify the errors in the weight perturbation at larger scales. Therefore, the cumulative $\mathbf{S}^4$ criterion becomes more advantageous as the number of edits increases.

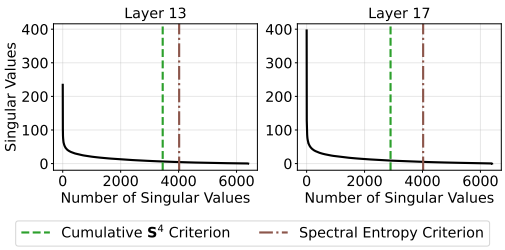

Figure 6: Comparison of the ranks retained by the cumulative $\mathbf{S}^4$ criterion and the spectral entropy criterion in the first and last edited layers of GPT-2 XL. The black line represents the singular values.

Table 11: Editing performance using the cumulative $\mathbf{S}^4$ criterion ($k_{\mathbf{S}^4}$) and the spectral entropy criterion (k_{erank}) on ZSRE and MQuAKE. N denotes the number of edits. Score denotes the harmonic mean of efficacy, generalization, and specificity. Eff., Gen., and Spec. denote efficacy, generalization, and specificity, respectively. The best results are highlighted in bold.

Model	Method	**ZSRE** (Score ↑)		
		$N = 100$	$N = 1,000$	$N = 2,000$
Llama 3	$k_{\mathbf{S}^4}$	**56.4**	**57.0**	**55.9**
	k_{erank}	56.2	56.9	55.4
GPT-J	$k_{\mathbf{S}^4}$	**53.9**	**54.0**	53.3
	k_{erank}	53.8	53.9	**53.4**
GPT-2 XL	$k_{\mathbf{S}^4}$	50.7	**46.2**	**45.5**
	k_{erank}	**50.8**	46.0	44.9

Model	Method	**ZSRE** ($N = 10,000$)		
		Eff.↑	Gen.↑	Spec.↑
GPT-2 XL	$k_{\mathbf{S}^4}$	**64.2**$_{\pm 1.0}$	**58.6**$_{\pm 0.8}$	**22.3**$_{\pm 0.3}$
	k_{erank}	57.1$_{\pm 0.9}$	51.4$_{\pm 0.8}$	19.4$_{\pm 0.1}$

Model	Method	**MQuAKE** (Accuracy ↑)		
		$N = 1,000$	$N = 2,000$	$N = 3,000$
GPT-2 XL	$k_{\mathbf{S}^4}$	**7.5**$_{\pm 0.5}$	**3.8**$_{\pm 0.4}$	**3.6**$_{\pm 0.3}$
	k_{erank}	**7.5**$_{\pm 0.3}$	3.2$_{\pm 0.3}$	2.0$_{\pm 0.1}$