# KELE: Residual Knowledge Erasure for Enhanced Multi-hop Reasoning in Knowledge Editing

**Anonymous ACL submission**


## Abstract

Large language models (LLMs) face challenges with internal knowledge inaccuracies and outdated information. Knowledge editing has emerged as a pivotal approach to mitigate these issues. Although current knowledge editing techniques exhibit promising performance in single-hop reasoning tasks, they show limitations when applied to multi-hop reasoning. Drawing on cognitive neuroscience and the operational mechanisms of LLMs, we hypothesize that the residual single-hop knowledge after editing causes edited models to revert to their original answers when processing multi-hop questions, thereby undermining their performance in multi-hop reasoning tasks. To validate this hypothesis, we conduct a series of experiments that empirically confirm our assumptions. Building on the validated hypothesis, we propose a novel knowledge editing method that incorporates a Knowledge Erasure mechanism for Large language model Editing (KELE). Specifically, we design an erasure function for residual knowledge and an injection function for new knowledge. Through joint optimization, we derive the optimal recall vector, which is subsequently utilized within a rank-one editing framework to update the parameters of targeted model layers. Extensive experiments on GPT-J (6B) and LLaMA-2 (7B) demonstrate that KELE substantially enhances the multi-hop reasoning capability of edited LLMs.


## 1 Introduction

Large Language Models (LLMs) have achieved significant success in a wide range of Natural Language Processing (NLP) tasks (Zhao et al., 2023). However, the knowledge embedded within LLMs can sometimes be factually incorrect or outdated, limiting their overall effectiveness. To address these limitations, knowledge editing methods have been proposed, offering a more efficient and precise approach to updating the knowledge in LLMs.

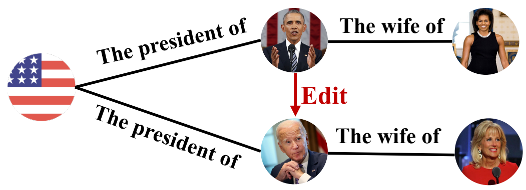



Figure 1: Example of Knowledge Editing

These methods have attracted considerable attention from researchers in recent years. Among these methods, those that modify the model's parameters are particularly important, as they provide a direct and flexible means of altering the model's behavior, such as KE (De Cao et al., 2021), (Mitchell et al., 2021), ROME (Meng et al., 2022a), and MEMIT (Meng et al., 2022b). This work focuses specifically on parameter-modifying approaches.

Although these editing methods have demonstrated promising results in single-hop reasoning evaluations, they still face significant challenges in multi-hop reasoning (Zhong et al., 2023). As illustrated in Figure 1, after editing the single-hop knowledge from "*The President of the USA is Obama*" to "*The President of the USA is Biden*," the edited model can easily answer the single-hop question, "*Who is the President of the USA?*" However, it struggles with multi-hop questions, such as "*Who is the wife of the President of the USA?*"

To better understand this challenge in knowledge editing for LLMs, we first analyze this problem from a cognitive neurological perspective. When the brain receives new information, it can activate neurons associated with related old memories, a phenomenon known as Memory Association (Roediger and McDermott, 1995; Schacter and Buckner, 1998; Kahana, 2012). This occurs because of the connectivity within neural networks, where the pathways of old memories are easily reactivated by relevant stimuli, thereby facilitating more efficient encoding and processing of new

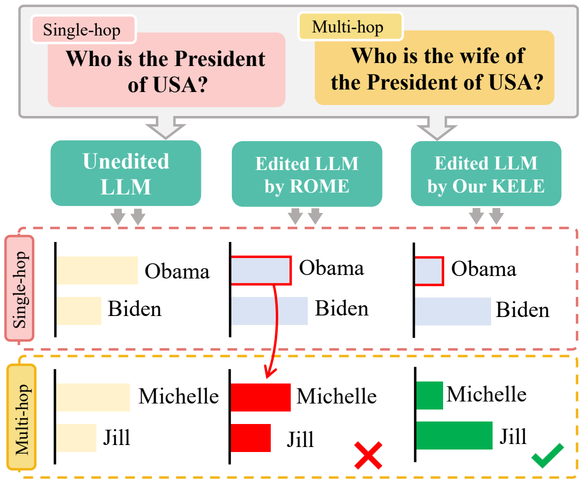



Figure 2: Single-hop and Multi-hop evaluation of Unedited LLM, LLM edited by ROME and our KELE. When confronted with a multi-hop question, the residual old single-hop knowledge (The President of the USA is Obama) in the LLMs edited by ROME prompts the model to generate the **original answer**, **Michelle** (Obama's wife), instead of the **correct answer**, **Jill** (Biden's wife).

information. LLMs exhibit a similar mechanism, where related knowledge stored in their parameters is activated and integrated during reasoning (Geva et al., 2021).

Building on these insights, we hypothesize the following reason for the poor performance of edited LLMs on multi-hop reasoning tasks: **LLMs retain a portion of single-hop old knowledge even after editing. When handling multi-hop questions related to the edited knowledge, the residual knowledge tends to prompt the models to produce original answers to these questions, thereby weakening their multi-hop reasoning ability.** For example, if the single-hop knowledge in the LLM is edited from "*The President of the USA is Obama*" to "*The President of the USA is Biden*," a portion of old knowledge "*The President of the USA is Obama*" may still be retained and reactivated within the model. As shown in Figure 2, when asked the multi-hop question "*Who is the wife of the President of the USA?*", the residual single-hop knowledge might cause the model to generate the original answer to the multi-hop question, *Michelle (Obama's wife)*, instead of the correct answer, *Jill (Biden's wife)*.

To verify this hypothesis, we investigate the relationship between the residual old knowledge in LLMs and their responses to multi-hop questions (Section 4). We define the Retain Score as a metric to quantify the residual old knowledge $(s, r, o)$ for each edit sample $(s, r, o, o*)$, utilizing the output logit score of $o$ under the prompt $p(s, r)$. As illustrated in Figure 3b, the higher the residual old knowledge in the edited LLM, the more likely it is to provide the original answers to multi-hop questions, resulting in a lower proportion of correct answers. Therefore, erasing the residual old knowledge offers a promising insight for improving the performance of edited LLMs on multi-hop reasoning tasks.

Based on the this hypothesis, we propose a simple yet effective method for large language model editing, termed Knowledge Erasure for Large Language Model Editing (KELE) (Section 5). Specifically, within the rank-one editing framework, we develop an old knowledge erasure function and a new knowledge injection function to jointly optimize and obtain the recall vector. This approach eliminates the interference of old knowledge while injecting new knowledge. Finally, the model parameters are updated in a single step using the recall vector and the subject representation through the rank-one update formula.

We summarize our contributions as follows:

- We investigate and validate the impact of residual old single-hop knowledge in edited LLMs on multi-hop reasoning tasks, demonstrating that such residual knowledge may cause edited LLMs to revert to original answers when faced with multi-hop questions.

- We integrate a knowledge erasure strategy into model editing and propose KELE, a simple yet effective editing method to enhance the multi-hop reasoning performance of edited LLMs.

- We conduct extensive experiments on LLaMA-2 (7B) and GPT-J (6B), showing that KELE significantly enhances the multi-hop reasoning ability of edited models.

## 2 Related Work

In this section, we review related research on knowledge editing and its challenges in multi-hop reasoning.

**Parameter-preserving Methods.** These methods typically store edited examples in an external knowledge base and guide the LLMs' output for specific queries by retrieving relevant knowledge. For instance, SERAC (Mitchell et al., 2022) employs a gating network along with an auxiliary model specifically designed to handle edited knowledge. T-patcher (Huang et al.) introduces extra trainable parameters in the last layer of the FFN to correct LLMs. However, these methods face a critical scalability issue: the complexity of managing the external model increases with each new edit, which may limit their practical usability.

**Parameter-modifying Methods.** These methods, including meta-learning, locat-and-edit, and fine-tuning-based approaches, edit LLMs by directly modifying their parameters. Meta-learning methods generate updated weights for LLMs by training a hyper-network. For example, KE (De Cao et al., 2021) uses a bi-directional LSTM to predict model weight updates, but it faced challenges with larger models due to their vast parameter spaces. To address this issue, MEND (Mitchell et al., 2021) employs a low-rank decomposition of fine-tuning gradients, providing an efficient mechanism for updating LLM weights. Locate-and-edit methods focus on identifying specific parameters associated with particular knowledge within LLMs, aiming for more interpretable and precise knowledge editing. Early efforts, such as KN (Dai et al., 2022), introduce a knowledge attribution method to identify knowledge neurons but struggles to precisely modify the model's weights. ROME (Meng et al., 2022a) method employs causal tracing to identify knowledge-relevant layers and then edits the corresponding FFN module. MEMIT (Meng et al., 2022b) further enhances this approach by improving the objective function and enabling multi-layer edits for batch editing. Recently, significant advancements in efficient parameter-tuning methods (Hu et al.; Ren et al., 2024) for supervised fine-tuning of LLMs have led to the development of fine-tuning-based editing methods (Ni et al., 2024; Gangadhar and Stratos, 2024), which utilizes LoRA (Hu et al.) and data augmentation strategies to directly fine-tune the LLMs, achieving the desired editing performance.

**Multi-hop reasoning in knowledge editing.** In recent years, several studies have aimed to enhance the performance of edited LLMs in multi-hop reasoning tasks. Zhong et al. (2023) introduce the MQuAKE dataset, specifically designed to evaluate the multi-hop reasoning capabilities of edited LLMs. They also propose a method that stores all edited facts externally, iteratively prompting LLMs to generate answers consistent with these edited facts. Building on this approach, PokeMQA (Gu et al., 2024) introduces auxiliary knowledge prompt to assist in question decomposition. GLAME (Zhang et al., 2024) leverages external knowledge graphs to capture the impact of target knowledge changes on high-order knowledge within LLMs. **These methods improve multi-hop reasoning by retrieving or incorporating external knowledge, which is not the focus of the current paper.** Additionally, Ju et al. (2024) find that LLMs often rely on factual shortcuts from pre-training corpora during reasoning, which contributes to the poor performance of edited models in multi-hop reasoning tasks. Unlike this study, we identify another potential cause for the poor performance of parameter-modified models in multi-hop reasoning tasks: the retention of old knowledge triggers the generation of original answers in multi-hop questions, thereby weakening the performance of edited models in these tasks. We validate this hypothesis through a series of experiments and propose a knowledge-erasure-based editing strategy to mitigate this issue.

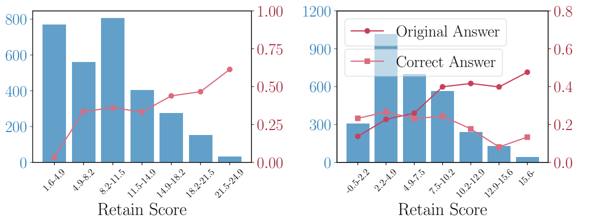
## 3 Preliminaries

In this section, we introduce the definition of knowledge editing and outline the corresponding tasks under single-hop and multi-hop evaluations.

**Definition 1. Knowledge Editing for LLMs** Knowledge editing (Yao et al., 2023) refers to the process of altering the behavior of an LLM $\mathcal{F}$'s to change encoded knowledge from $(s, r, o)$ to the new knowledge $(s, r, o^*)$. Here, knowledge is represented as a triple, with $s$ as the subject, $r$ as the relation, and $o$ as the object. Each editing instance $e$ is denoted as $(s, r, o, o^*)$, and the LLM after editing is referred to as $\mathcal{F}'$.

**Definition 2. Single-hop Evaluation in Knowledge Editing** Single-hop evaluation assesses whether an edit $(s, r, o, o^*)$ is successful in an edited LLM $\mathcal{F}'$. This evaluation constructs prompts $p(s, r)$ based on the subject $s$ and relation $r$, and measures the performance of $\mathcal{F}'$ using Efficacy, Paraphrase and Specificity metrics (Yao et al., 2023).

**Definition 3. Multi-hop Evaluation in Knowledge Editing** Multi-hop evaluation examines whether the edited LLMs can effectively utilize the updated knowledge for reasoning in multi-hop tasks. Given a chain of facts $(s_1, r_1, o_1), ..., (s_n, r_n, o_n)$, where the object of the $i$-th fact also serves as the subject of the next fact in the chain, i.e., $o_i = s_{i+1}$, a multi-hop question




(a) Unedited GPT-J (b) Edited GPT-J

Figure 3: (a) The accuracy of single-hop answer generated by unedited GPT-J . (b)The accuracy of original and correct answers generated by edited GPT-J. The left y-axis represents the number of instances within each Retain Score interval, while the right y-axis indicates the accuracy.

$p(s_1, r_1, .., r_n)$ can be constructed, with the answer being $o_n$. For example, with a chain consisting of two facts, *(USA, president of, Obama)* and *(Obama, wife of, Michelle)*, one can write a 2-hop question: *Who is the wife of the president of USA?*. Once one or more facts in the chain are edited, e.g., *(USA, president of, Obama)* is edited to *(USA, president of, Biden)*, the edited LLM must utilize the new knowledge to answer the multi-hop question. The model's response should change from the original answer *Michelle* to the correct answer *Jill*.

## 4 Analysis of the Impact of Old Knowledge on Multi-hop Reasoning

In this section, we validate our hypothesis by examining the impact of old knowledge on the performance of edited LLMs in multi-hop reasoning. We select the representative multi-hop reasoning evaluation dataset, MQuAKE (Zhong et al., 2023), to conduct experiments. Each instance in MQuAKE is represented as $d = (\mathcal{E}, \mathcal{Q}, a, a^*)$. Here, $\mathcal{E}$ denotes the set of single-hop edits $e = (s, r, o, o^*)$, $\mathcal{Q}$ represents a multi-hop question evaluating editing performance, and $a$ and $a^*$ correspond to the original and correct answers to $\mathcal{Q}$. Further details on MQuAKE are provided in Section 6.1 and Appendix B.

### 4.1 Retain Score

We first define a metric to quantify the retention of old knowledge in the LLM. In cognitive neuroscience, memory activation is often measured by the intensity of neural activity. Analogously, in LLMs, the logit vector can serve as an indicator of the model's memory activation strength. Building on this concept, we introduce the **Retain Score (RS)** indicator for each edit sample $e = (s, r, o, o^*)$ to measure the residual presence of the old knowledge $(s, r, o)$.

When an LLM is given an input prompt, it generates the next token based on the logit vector produced by its final layer. A higher logit value for a token indicates greater model confidence in generating that token, corresponding to stronger memory activation. Consequently, we use the logit value as a measure of the model's retention of old knowledge. To ensure a consistent assessment of retention across different editing instances, we standardize the logit vectors to eliminate variations from varying logit distributions:

$$\mathbf{RS}(e) = \frac{D_o - \mu}{\sigma}, \tag{1}$$

where $D$ represents the logit vector produced by the final layer of the LLM, $D_o$ is the logit score of $o$, while $\mu$ and $\sigma$ denote the mean and standard deviation of the logit vector $D$, respectively.

#### 4.1.1 The reasonableness of Retain Score

To validate the reasonableness of the Retain Score, we first divide the RS values of all edit samples in the dataset into different intervals. For each interval, we then calculate the probability that the unedited model correctly answers $o$ given the prompt $p(s, r)$. The experimental results, as shown in Figure 3a, indicate that as the RS value increases, the accuracy of the unedited model's responses also increases. This suggests that the model's sensitivity to the corresponding knowledge strengthens as the RS value rises, demonstrating that the RS metric effectively measures the retention of old knowledge.

### 4.2 Impact of Old Knowledge on Multi-hop Reasoning

To further investigate the impact of residual old knowledge on multi-hop reasoning, we apply the ROME method to GPT-J and explore the relationship between the Retain Score and the accuracy of answering multi-hop questions.

Specifically, for each instance $d$, we first calculate the accumulated old single-hop knowledge of all edit samples $\mathcal{E}$ in the edited models:

$$\mathbf{RS}(d) = \sum_{e \in \mathcal{E}} \mathbf{RS}(e). \tag{2}$$

We then divide the dataset into different subsets based on the varying ranges of the Retain Score of the instances. For each subset, we calculate the accuracy of the edited model in answering the original and correct answers to the multi-hop questions. The results are shown in Figure 3b.

As illustrated in Figure 3b, we observe that as the Retain Score value increases, the edited models show a significant improvement in accuracy when providing the original answers to multi-hop questions. However, the accuracy of the edited model in providing correct answers decreases as the Retain Score rises. This suggests that as the amount of retained old knowledge increases, the model becomes more likely to favor the original answers, thereby diminishing its ability to generate correct responses to multi-hop questions.

These experiments validate that **LLMs retain traces of old single-hop knowledge after editing, which significantly motivates them to revert to original answers for multi-hop questions and undermines their performance in providing correct answers.** Therefore, eliminating residual old knowledge during the editing process is crucial for enhancing the accuracy of LLMs in multi-hop reasoning.

## 5 Methodology

In this section, we introduce the proposed KELE, with its architecture depicted in Figure 4. The KELE framework integrates a knowledge erasure strategy within the rank-one model editing framework (Meng et al., 2022a). Specifically, KELE targets a specific layer $l$ and transforms knowledge editing into two key operations: old knowledge erasure and new knowledge injection, which together are used to compute the recall vector $\mathbf{v}_*$. Subsequently, $\mathbf{v}_*$, along with the subject representation $\mathbf{k}_*$, is applied in Equation (11) to update the parameters of the second layer of the FNN, thereby completing the knowledge editing process.

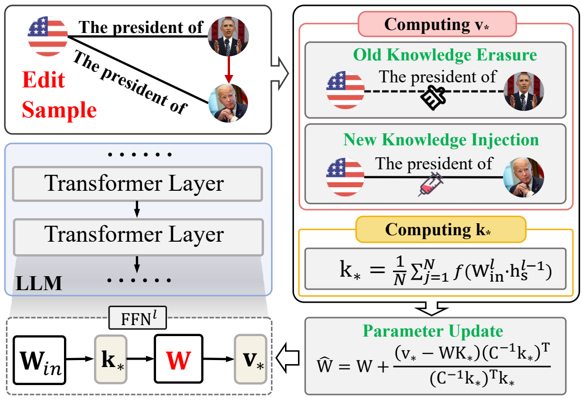



Figure 4: Overview of KELE architecture. First, we use the old knowledge erasure function and the new knowledge injection function to derive the recall vector $\mathbf{v}_*$. Then, we compute the subject representation $\mathbf{k}_*$. Finally, the parameters are updated using the rank-one update formula.

### 5.1 Computing $\mathbf{v}_*$ to Recall New Knowledge

To effectively edit new knowledge while minimizing the negative impact of old knowledge on multi-hop reasoning, we construct an old knowledge erasure function and a new knowledge inject function, which are jointly optimized to obtain $\mathbf{v}_*$. In this process, we optimize the learnable parameter vector $\mathbf{h}$ to modify the original value vector $\mathbf{v}_s^l$, resulting in the optimal vector $\mathbf{v}_* = \mathbf{v}_s^l + \mathbf{h}$.

#### 5.1.1 Old knowledge erasure function

To mitigate the influence of residual old knowledge that may still prompt the edited LLM to generate the original answer in response to multi-hop question, we define a margin-based erasure loss for calculating $\mathbf{v}_*$. Specifically, given an edit sample $(s, r, o, o^*)$, we aim to suppress the logit score of $o$ in the output distribution when responding to the query $p(s, r)$. Let $D = \mathrm{LLM}(p(s, r); \mathbf{v}_s^l + \mathbf{h})$ be the output logit vector obtained by modifying the subject token's hidden state $\mathbf{v}_s^l$ via a learnable perturbation vector $\mathbf{h}$. The old knowledge erasure loss is then defined as:

$$\mathcal{L}_e = \max\left(0, D_o - D_{[k]}\right), \tag{3}$$

where $D_o$ is the logit score of $o$, and $D_{[k]}$ is the $k$-th highest logit value in $D$. This formulation penalizes the model only if $o$ remains among the top-$k$ predictions, thus avoiding unnecessary suppression that may lead to collateral damage.

#### 5.1.2 New knowledge injection function

For each edit sample $(s, r, o, o^*)$, our second objective is to refine the parameter vector $\mathbf{h}$ enables the LLM to accurately predict the target object $o^*$. Accordingly, the knowledge injection loss function is defined as:

$$\mathcal{L}_p = -\frac{1}{N}\sum_{j=1}^{N} \log \mathrm{P}_{\mathcal{F}(\mathbf{v}_s^l += \mathbf{h})}[o^* \mid x_j \oplus p(s, r)], \tag{4}$$

where $x_j$ is the random prefix generated by the LLM to foster optimization robustness, and $\mathcal{F}(\mathbf{v}_s^l += \mathbf{h})$ indicates the LLM's inference alteration through the hidden state $\mathbf{v}_s^l$ modification to $\mathbf{v}_s^l + \mathbf{h}$.

To mitigate the impact of above operations on the intrinsic of $s$ within the LLM, we minimize the KL divergence between $\mathcal{F}(\mathbf{v}_s^l += \mathbf{h})$ and the original model $\mathcal{F}$ (Meng et al., 2022a):

$$\mathcal{L}_a = D_{\mathrm{KL}}\left(\mathrm{P}_{\mathcal{F}(\mathbf{v}_s^l += \mathbf{h})}[x \mid p'] \parallel \mathrm{P}_{\mathcal{F}}[x \mid p']\right), \tag{5}$$

where $p'$ denotes prompts in the form of "subject is a".

Ultimately, the parameter $\mathbf{h}$ is optimized by minimizing the following objective function:

$$\mathcal{L} = \mathcal{L}_e + \mathcal{L}_p + \lambda \mathcal{L}_a, \tag{6}$$

where $\lambda$ adjusts the regularization strength. Throughout the optimization process, the parameters of the LLM remain unchanged.

### 5.2 Computing $\mathbf{k}_*$ to Represent Subject

For each edit sample $(s, r, o, o^*)$, the subject representation $\mathbf{k}_*$ is calculated by

$$\mathbf{k}_* = \frac{1}{N}\sum_{j=1}^{N} f(\mathbf{W}_{in}^l \cdot \mathbf{h}_s^{l-1}). \tag{7}$$

Here, we also utilize $N$ random prefixes generated in the same manner as for the computing $\mathbf{v}_*$ (Meng et al., 2022a).

After obtaining the optimized vectors $\mathbf{v}_*$ and $\mathbf{k}_*$, we substitute them into the following equation to get the updated parameter $\hat{\mathbf{W}}$. The detailed procedure in provided in Appendix A:

$$\hat{\mathbf{W}} = \mathbf{W} + \frac{(\mathbf{v}_* - \mathbf{W}\mathbf{k}_*)(\mathbf{C}^{-1}\mathbf{k}_*)^{\mathrm{T}}}{(\mathbf{C}^{-1}\mathbf{k}_*)^{\mathrm{T}}\mathbf{k}_*}. \tag{8}$$

## 6 Experiments

In this section, we evaluate our KELE by applying it to two datasets and assessing its performance on two auto-regressive LLMs. We aim to answer the following questions through experiments.

- **Q1**: How does KELE perform in multi-hop and single-hop reasoning evaluation compared with state-of-the-art editing methods?
- **Q2**: How does the degree of erasure of old knowledge affect model's performance in multi-hop reasoning?
- **Q3**: What impact does our KELE have on the retention of old knowledge?

### 6.1 Experimental Setups

#### 6.1.1 Datasets and Evaluation Metrics

We evaluate our KELE on two representative datasets:**MQuAKE-3K** (Zhong et al., 2023) and **CounterFact** (Meng et al., 2022a). Detailed descriptions of the datasets and evaluation metrics are provided in Appendix B and C.

**MQuAKE-3K** is a challenging dataset designed to assess models' ability to perform multi-hop reasoning using newly edited knowledge. Each entry in this dataset involves multiple single-hop edits and includes multi-hop reasoning questions. This imposes stricter demands on the capability of edited LLMs to utilize the updated knowledge. Following (Zhong et al., 2023), we use *Multi-hop Accuracy* to measure the performance of edited LLMs. To fully leverage the LLM's reasoning ability, we employ three approaches when generating answers: Zero-shot, Few-shot, and Chain-of-Thought (CoT). The details of prompting are shown in Appendix F.

**CounterFact** is a dataset focused on evaluating LLMs' ability to recall edited knowledge in a single-hop setting, as well as to assess the impact of editing operations on unrelated knowledge within the LLMs. Following (Meng et al., 2022a), we employ three widely used metrics for this dataset: *Efficacy Score*, which measures the success rate of edits; *Paraphrase Score*, which evaluates the model's ability to accurately recall edited knowledge in paraphrased forms, testing its generalization ability; and *Neighborhood Score*, which assesses whether irrelevant knowledge in the LLM is disturbed.

#### 6.1.2 Baselines

We conduct experiments on LLaMA-2 (7B) (Touvron et al., 2023) and GPT-J (6B) (Wang and Komatsuzaki, 2021). Since our study focuses on the impact of residual old knowledge in parameter-modification-based methods, we compare our approach against representative baselines in this category: Constrained Fine-Tuning (FT) (Zhu et al., 2020), MEND (Mitchell et al., 2021), ROME (Meng et al., 2022a), and MEMIT (Meng et al., 2022b). Implementation details for both baselines and KELE are provided in Appendix D and E.

### 6.2 Performance Comparison (RQ1)

The performance of all editors on the MQuAKE-3K and CounterFact is presented in Tables 1 and 2. Figure 5 provides a comprehensive com-

| Editor | Correct Answer ↑ | | | | Original Answer ↓ | | | |
|---|---|---|---|---|---|---|---|---|
| | Average Accuracy | Zero-Shot | Few-Shot | CoT | Average Accuracy | Zero-Shot | Few-Shot | CoT |
| LlaMA2 | 7.73 | 5.27 | 11.70 | 6.23 | 48.44 | 35.93 | 42.63 | 66.77 |
| FT | 13.41 | 7.90 | 17.57 | 14.77 | 45.64 | 32.73 | 41.20 | 63.00 |
| MEND | 12.67 | 8.20 | 15.00 | 14.80 | 45.43 | 33.23 | 41.43 | 61.63 |
| ROME | 13.99 | 8.33 | 15.87 | 17.77 | 37.26 | 27.20 | 34.10 | 50.47 |
| MEMIT | 17.14 | 8.63 | 15.57 | 27.23 | 33.75 | 23.00 | 34.43 | 43.83 |
| **KELE** | **19.01** | **9.90** | **18.12** | **29.01** | **27.43** | **19.61** | **28.11** | **34.57** |
| $\Delta Improve$ | 10.91% | 14.72% | 3.13% | 6.54% | 18.73% | 14.74% | 17.57% | 21.13% |
| GPT-J | 5.47 | 2.91 | 4.58 | 8.92 | 35.22 | 28.16 | 22.01 | 55.48 |
| FT | 6.94 | 3.79 | 5.55 | 11.47 | 33.27 | 26.07 | 20.34 | 53.40 |
| MEND | 11.17 | 4.37 | 6.70 | 22.43 | 29.40 | 24.77 | 17.37 | 46.07 |
| ROME | 14.56 | 7.54 | 8.69 | 27.46 | 18.40 | 12.85 | 13.64 | 28.71 |
| MEMIT | 9.09 | 3.74 | 5.46 | 18.07 | 27.35 | 19.69 | 19.42 | 42.95 |
| **KELE** | **16.36** | **9.12** | **10.08** | **29.87** | **13.28** | **10.44** | **11.49** | **17.90** |
| $\Delta Improve$ | 12.36% | 20.95% | 16.00% | 8.78% | 27.83% | 18.75% | 15.76% | 37.65% |

Table 1: Performance comparison of editors on multi-hop questions of MQuAKE-3K dataset in terms of Multi-hop Accuracy (%).↑ indicates that higher values correspond to better performance, while ↓ indicates that lower values correspond to better performance.

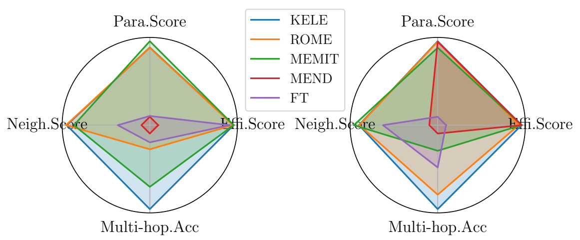



Figure 5: Comparative performance on LLaMA-2 (left) and GPT-J (right) across different metrics.

parison of all editing methods across four metrics on both datasets, demonstrating that KELE exhibits relatively balanced and superior performance across all metrics, particularly excelling in Multi-hop Accuracy, where it significantly outperforms other methods.

**Results on MQuAKE-3K** As shown in Table 1, our KELE outperforms all baselines by a significant margin across all evaluation metrics and settings. Specifically, KELE demonstrates improvements of 10.91 % and 12.36 % in average multi-hop accuracy over the best baseline models for LLaMA-2 and GPT-J, respectively. This indicates that KELE effectively enhances the ability of edited LLM in multi-hop reasoning tasks. Additionally, the multi-hop accuracy of KELE in generating original answers decreased by an average of 18.73% and 27.83 % on LLaMA-2 and GPT-J, respectively, compared to the strongest baseline model. This suggests that the knowledge erasure operations in KELE successfully mitigate the recall of old knowledge in the edited LLMs when performing complex reasoning tasks. These findings further support our hypothesis that residual old knowledge in the edited models is easily recalled during multi-hop reasoning. This recall causes the model to produce original answers to multi-hop questions, thereby weakening the LLM's performance on this task.

**Results on CounterFact** Unlike the MQuAKE-3K dataset, which primarily evaluates multi-hop reasoning, the CounterFact dataset focuses on assessing the single-hop factual recall of edited knowledge. From Table 2, we observe a clear trade-off between Para.Score and Neigh.Score across different methods and architectures. Specifically, KELE achieves the best or near-best Neigh.Score on LLaMA-2 and Para.Score on GPT-J, while also maintaining consistently high Effi.Score. Although KELE yields a lower Neigh.Score than MEMIT on GPT-J, it significantly outperforms MEMIT in Paraphrase Score. Conversely, on LLaMA-2, KELE performs slightly worse in Para.Score but better in Neigh.Score compared to MEMIT. Across both architectures, the overall average performance of KELE remains competitive with ROME and MEMIT. A detailed analysis of the potential side effects introduced by the knowledge erasure mechanism is provided in Appendix H. Nonetheless, given the substantial improvements observed in multi-hop reasoning tasks, we consider this trade-off to be both reasonable and acceptable.

| Editor | Effi.Score | Para.Score | Neigh.Score | Avg. |
|---|---|---|---|---|
| LlaMA2 | 13.7 | 16.65 | 83.4 | 20.68 |
| FT | 99.60 | 55.08 | 68.80 | 70.21 |
| MEND | 92.85 | 54.65 | 62.83 | 66.69 |
| ROME | 99.95 | 92.62 | **81.87** | 90.98 |
| MEMIT | **100** | **96.22** | 79.86 | **91.14** |
| KELE | **100** | 92.78 | 81.68 | 90.85 |
| GPT-J | 16.30 | 18.60 | 83.00 | 23.59 |
| FT | **100** | 98.80 | 10.30 | 25.60 |
| MEND | 97.40 | 53.60 | 53.90 | 63.19 |
| ROME | 99.90 | 98.88 | 76.02 | 90.15 |
| MEMIT | **100** | 95.23 | **81.26** | **91.44** |
| KELE | 99.90 | **99.15** | 76.39 | 90.40 |

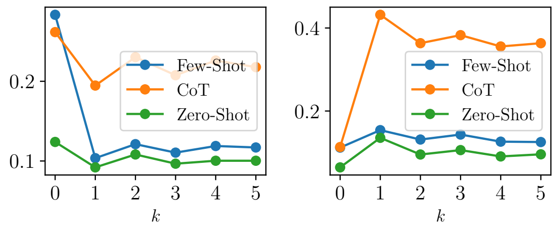
Table 2: Performance comparison on COUNTERFACT in terms of Efficacy Score (%), Paraphrase Score (%), and Neighborhood Score (%). The Avg. (%) is the harmonic mean of the three evaluation metrics. The best performance is highlighted in boldface, and the second-best is underlined. Gray numbers indicate a clear failure on the metric.

### 6.3 Impact of Erasure Internsity (RQ2)

The hyperparameter $k$ of Equation (3) represents the degree of erasure of old knowledge. A larger $k$ indicates a higher degree of erasure, and vice versa. To investigate the impact of varying erasure intensities on the model, we conduct experiments with various $k$ values on the MQUAKE-3K dataset. The results, shown in Figure 6, lead to the following observations: As $k$ increases, the erasure of old knowledge is enhanced, and the accuracy of generating original answers for multi-hop questions gradually decreases. This further validates that residual old knowledge after editing encourages models to revert to original answers in multi-hop questions. Furthermore, the edited GPT-J achieve its best performance at $k = 1$, with the highest accuracy in generating correct answers. Beyond this point, as $k$ continues to increase, the performance of the models either stabilizes or declines. This may be due to overly high erasure intensity. While it reduces the likelihood of generating original answers, it may also introduce other disruptions to the model, ultimately weakening its reasoning ability.

### 6.4 The impact on Old Knowledge (RQ3)

To investigate the impact of KELE on old knowledge $(s, r, o)$, we examine the distribution of Retain Score in three models: the unedited LLM (GPT-J), the LLM edited with ROME, and the LLM edited with KELE. The experimental results are presented in Figure 7. From the results, we observe that the unedited model exhibits the highest Retain Scores,




(a) Original Answer (b) Correct Answer

Figure 6: Performance of edited GPT-J with different $k$.

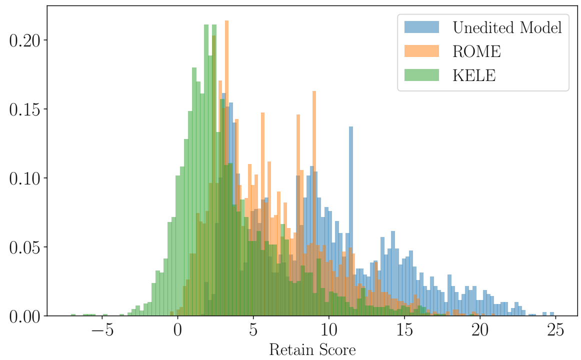



Figure 7: The distribution of Retain Score.

with a significant density around 10 to 15, indicating substantial retention of old knowledge. The ROME-edited model shows a reduction in Retain Score, shifting the distribution leftward, but still retains a noticeable amount of old knowledge, particularly in the 5 to 10 range. In contrast, the KELE demonstrates the most significant reduction, with a peak near lower Retain Scores. These results demonstrate that KELE effectively erases residual old knowledge, which is crucial for enhancing the model's performance in multi-hop reasoning tasks.

## 7 Conclusion

In this paper, we identify that the poor performance of current parameter-modifying editing methods in multi-hop scenarios stems from the retention of single-hop old knowledge, which leads LLMs to revert to original answers. Inspired by neuroscience, we propose KELE, a simple yet effective method that integrates a knowledge erasure mechanism into a rank-one model editing framework. By jointly erasing outdated knowledge and injecting new facts, KELE significantly improves multi-hop reasoning. Experiments on two LLMs, along with detailed analysis, validate its effectiveness and superiority.

## Limitations

Despite the effectiveness of our approach, there are several limitations that warrant further exploration.

First, the hyperparameter $k$ in Equation (3), which controls the strength of knowledge erasure, is dependent on manual selection. Different values of $k$ lead to varying degrees of erasure, making it challenging to determine the optimal setting across different scenarios. A promising direction for future work is to develop an adaptive erasure strategy that dynamically adjusts based on the amount of residual old knowledge, ensuring a balance between effective editing and minimal unintended interference.

Second, while the erasure of residual knowledge significantly improves multi-hop reasoning and maintains competitive overall editing performance, it may also introduce unintended side effects, as discussed in Appendix H. Future work will explore more refined erasure strategies to further reduce interference with unrelated knowledge.

## Ethical Considerations

We realize that there are risks in developing generative LLMs, so it is necessary to pay attention to the ethical issues of LLMs. We use publicly available pre-trained LLMs, i.e., LLaMA-2 (7B) and GPT-J (6B). The datasets are publicly available, i.e., COUNTERFACT and MQUAKE. All models and datasets are carefully processed by their publishers to ensure that there are no ethical problems.

## References

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Annual Meeting of the Association for Computational Linguistics*, pages 8493–8502.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506.

Govind Gangadhar and Karl Stratos. 2024. Model editing by pure fine-tuning. *arXiv preprint arXiv:2402.11078*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Hengrui Gu, Kaixiong Zhou, Xiaotian Han, Ninghao Liu, Ruobing Wang, and Xin Wang. 2024. PokeMQA: Programmable knowledge editing for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8069–8083. Association for Computational Linguistics.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations*.

Tianjie Ju, Yijin Chen, Xinwei Yuan, Zhuosheng Zhang, Wei Du, Yubin Zheng, and Gongshen Liu. 2024. Investigating multi-hop factual shortcuts in knowledge editing of large language models. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Michael Jacob Kahana. 2012. *Foundations of human memory*. OUP USA.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2023. Feed-forward blocks control contextualization in masked language models. *arXiv preprint arXiv:2302.00456*.

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

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. In *International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. In *International Conference on Learning Representations*.

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

Shiwen Ni, Dingwei Chen, Chengming Li, Xiping Hu, Ruifeng Xu, and Min Yang. 2024. Forgetting before learning: Utilizing parametric arithmetic for knowledge updating in large language models. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Judea Pearl. 2022. Direct and indirect effects. In *Probabilistic and causal inference: the works of Judea Pearl*, pages 373–392.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pengjie Ren, Chengshun Shi, Shiguang Wu, Mengqi Zhang, Zhaochun Ren, Maarten de Rijke, Zhumin Chen, and Jiahuan Pei. 2024. MELoRA: Mini-ensemble low-rank adapters for parameter-efficient fine-tuning. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Henry L Roediger and Kathleen B McDermott. 1995. Creating false memories: Remembering words not presented in lists. *Journal of experimental psychology: Learning, Memory, and Cognition*, 21(4):803.

Daniel L Schacter and Randy L Buckner. 1998. Priming and the brain. *Neuron*, 20(2):185–195.

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

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. *Annual Conference on Neural Information Processing Systems*, 33:12388–12401.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. In *Conference on Empirical Methods in Natural Language Processing*, pages 10222–10240.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. 2024. Knowledge graph enhanced large language model editing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22647–22662.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Conference on Empirical Methods in Natural Language Processing*, page 15686–15702.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

# A Rank-One Model Editing

Rank-One Model Editing (ROME) (Meng et al., 2022a) is a Locate-then-edit method that presupposes factual knowledge is stored within the Feedforward Neural Networks (FFNs), conceptualized as key-value memories (Geva et al., 2021; Kobayashi et al., 2023). The output of the $l$-th layer FFN for the $i$-th token is given by:

$$\mathbf{v}_i^l = f(\mathbf{W}_{in}^l \cdot \mathbf{h}_i^{l-1}) \cdot \mathbf{W}^l, \tag{9}$$

where $f(\cdot)$ denotes the activation function, and $\mathbf{h}_i^{l-1}$ is the FFN input. For simplicity, the superscript $l$ is omitted in the following discussion.

In this context, $f(\mathbf{W}_{in} \cdot \mathbf{h}_i)$ functions as the keys, denoted as $\mathbf{k}_i$. The outputs of the subsequent layer represent the corresponding values. Utilizing casual tracing (Pearl, 2022; Vig et al., 2020), this method identify a specific FFN layer for editing and updates the weight $\mathbf{W}$ of the second layer by solving a constrained least-squares problem:

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{WK} - \mathbf{V}\|, \\ \text{subject to} \quad & \mathbf{Wk}_* = \mathbf{v}_*. \end{aligned} \tag{10}$$

where the objective function aims to preserve the knowledge unrelated to the edited sample within the LLM. Here, $\mathbf{K} = [\mathbf{k}_1; \mathbf{k}_2; , \ldots, ; \mathbf{k}_p]$ denotes the sets of keys encoding subjects unrelated to the edited fact, and $\mathbf{V} = [\mathbf{v}_1; \mathbf{v}_2; , \ldots, ; \mathbf{v}_p]$ represents the corresponding values. The constraint ensures that the edited knowledge is incorporated into the FFN layer by enabling the key $\mathbf{k}_*$ (encoding subject $s$) to retrieve the value $\mathbf{v}_*$ about the new object $o^*$.

As explicated in (Meng et al., 2022a), a closed-form solution to the optimization problem can be derived:

$$\hat{\mathbf{W}} = \mathbf{W} + \frac{(\mathbf{v}_* - \mathbf{Wk}_*)(\mathbf{C}^{-1}\mathbf{k}_*)^{\mathrm{T}}}{(\mathbf{C}^{-1}\mathbf{k}_*)^{\mathrm{T}}\mathbf{k}_*}, \tag{11}$$

where $\mathbf{C} = \mathbf{KK}^{\mathrm{T}}$ is a constant matrix, precomputed by estimating the uncentered covariance of $\mathbf{k}$ based on a sample of Wikipedia text (Appendix E). Thus, solving the optimal parameter $\hat{\mathbf{W}}$ is transformed into calculating subject representation $\mathbf{k}_*$ and recall vector $\mathbf{v}_*$.

# B Dataset

We evaluate our KELE on two representative datasets:**MQuAKE-3K** (Zhong et al., 2023) and **CounterFact** (Meng et al., 2022a).

### B.1 Details of MQuAKE-3K Dataset

MQuAKE-3K is a challenging dataset designed to assess models' ability to perform multi-hop reasoning using newly edited knowledge. Each entry in this dataset involve multiple edits and includes multi-hop reasoning questions that require reasoning from 2 to 4 hops to answer correctly. This imposes stricter demands on the capability of edited LLMs to utilize the updated knowledge. Table 3 provides an example from MQuAKE-3K dataset. In this example, two edits are required: inserting the knowledge *(Lou Pearlman, is a citizen of, India)* and *(India, The capital of, Taloga)*. Accordingly, a 3-hop question "*What is the capital of the country to which Lou Pearlman belonged?*" is constructed to assess the post-edit models's ability to ulitze edited knowledge and its related information. Following (Zhong et al., 2023), our evaluation focuses on a subset of 3000 entries, evenly distributed across $\{2, 3, 4\}$-hop questions, with each category comprising $1000$ entries.

### B.2 Details of CounterFact Dataset

Table 4 presents an example from the CounterFact dataset. Each entry includes an edit request, several paraphrase prompts, and neighborhood prompts. In this example, the edit request aims to change the model's knowledge of *The mother tongue of Go Hyeon-jeong* from *Korean* to *French*. Paraphrase prompts are semantic variations of the target prompt, while neighborhood prompts involve the same relation but with a different subject, whose knowledge should remain unaffected by the edit. Our train/test dataset splits are kept the same as (Meng et al., 2022a). Similarly, we evaluate our method using the first 2000 records on GPT-J and LLaMA-2.

## C Evaluation Metrics

For each instance $d = (\mathcal{E}, \mathcal{Q}, a, a^*)$ in the MQuAKE dataset, the multi-hop accuracy after editing is defined as:

$$\frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbb{I}\left[\mathcal{F}'(q) = a^*)\right].$$

We report the averaged multi-hop accuracy in our evaluation.

For the CounterFact dataset, we use three widely-used metrics (Meng et al., 2022a,b), Efficacy Score, Paraphrase Score, and Neighborhood Score to evaluate all editors. Each metric is calculated as follows:

**Efficacy Score** is to test whether the post-edit LLMs can correctly recall the new target entity when given the edit prompt $p(s, r)$. It is calculated by

$$\mathbb{E}\left[\mathbb{I}\left[\mathrm{P}_{\mathcal{F}'}\left(o^* \mid p(s,r)\right) > \mathrm{P}_{\mathcal{F}'}\left(o \mid p(s,r)\right)\right]\right].$$

**Paraphrase Score** measures the performance of the post-edit LLM on rephase prompt set $P^P$ of edit prompt $p(s, r)$. The calculation is similar to the Efficacy Score:

$$\mathbb{E}_{p \in P^P}\left[\mathbb{I}\left[\mathrm{P}_{\mathcal{F}'}\left(o^* \mid p\right) > \mathrm{P}_{\mathcal{F}'}\left(o \mid p\right)\right]\right].$$

**Neighborhood Score** measures whether the post-edit LLM assigns the higher probability to the correct fact on the prompt set $P^N$, which consists of distinct but semantically similar prompts $p(s, r)$. The calculation is defined as:

$$\mathbb{E}_{p \in P^N}\left[\mathbb{I}\left[\mathrm{P}_{\mathcal{F}'}\left(o^* \mid p\right) < \mathrm{P}_{\mathcal{F}'}\left(o \mid p\right)\right]\right].$$

This metric can assess the extent of the impact that edits have on unrelated knowledge.

## D Baselines

Our experiments are conducted on LLaMA-2 (7B) (Radford et al., 2019) and GPT-J (6B) (Wang and Komatsuzaki, 2021), and we compare KELE with the following state-of-the-art editing methods:

**Constrained Fine-Tuning (FT)** (Zhu et al., 2020) involves fine-tuning specific layers of the LLM's parameters directly using gradient descent, while imposing a norm constraint on the weight changes to prevent catastrophic forgetting.

**MEND** (Mitchell et al., 2021) utilizes a hypernetwork based on the low-rank decomposition of gradients to perform editing.

**ROME** (Meng et al., 2022a) is based on the hypothesis that knowledge in LLMs is stored in the FFN module, and uses optimization to update a FFN layer to insert knowledge.

**MEMIT** (Meng et al., 2022b) builds on the ROME method, specializing in batch-editing tasks by performing edits on a range of FFN layers.

## E Implementation Details

We implement our KELE method using **PyTorch**[1]. For the other baselines, we conduct our experiments using the code provided by ROME (Meng

[1] https://pytorch.org/

| Property | Value |
|---|---|
| Edit Request 1 | {Lou Pearlman } is a citizen of *United States of America* → *India* |
| Edit Request 2 | The capital of {India} is *New Delhi* → *Taloga* |
| New Question | What is the capital of the country to which Lou Pearlman belonged? |
| Original Relation | (Lou Pearlman, a citizen of, United States of America), (United States of America, the capital of, Washington) |
| Original Answer | Washington |
| New Relation | (Lou Pearlman, a citizen of, India), (India, the capital of, Taloga) |
| New Answer | Taloga |

Table 3: An Example of MQUAKE dataset

| Property | Value |
|---|---|
| Edit Request | The mother tongue of {Go Hyeon-jeong} is *Korean* → *French* |
| Efficacy_prompt | The mother tongue of Go Hyeon-jeong is |
| Paraphrase_prompt | It won the Governor General's Literary Award the same year. Go Hyeon-jeong spoke the language |
| Neighborhood_prompt | The native language of Gong Ji-young is |

Table 4: An Example of COUNTERFACT dataset

et al., 2022a), ensuring that all settings, including hyperparameters, are consistent with (Meng et al., 2022a,b). For our KELE, editing operation is performed at layer 7 for GPT-J with the optimal $k$ value of 3, selected after searching within $k = \{0, 1, 2, 3, 4, 5\}$. For LLaMA-2, editing is carried out at layer 5, and the optimal $k$ value of 3 chosen from the same search space. Other parameters are kept consistent with those used in ROME. We run the evaluation five times with different random seeds and report the mean value of each method. Our experiments are conducted on NVIDIA Tesla A100 (80G) and AMD EPYC 7742 CPU. On LLaMA-2, editing takes 39s per sample on average, with 33,746.0 MiB GPU memory usage. On GPT-J, editing takes 24s per sample on average, with 31,936.0 MiB GPU memory usage.

## F Prompt used in MQUAKE

To fully leverage the LLM's reasoning ability, we employ three approaches when generating answers: Zero-shot, Few-shot, and Chain-of-Thought (CoT). The templates of few-shot prompt and CoT prompt are shown in Figures 8 and 9.

## G Impact of Erasure Internsity

Figure 10 shows the performance of LLaMA-2 with various $k$ values on the MQUAKE-3K dataset. The results indicate that as as $k$ increases, the erasure of old knowledge is enhanced, and the accuracy of generating original answers for multi-hop questions gradually decreases. This further validates that residual old knowledge after editing encourages models to revert to original answers in multi-hop questions. Furthermore, the edited LLaMA-2 achieves its best performance at $k = 4$, with the highest accuracy in generating correct answers. Beyond this point, as $k$ continues to increase, the performance of the models either stabilizes or declines. This may be due to excessively high erasure intensity. While it reduces the likelihood of generating original answers, it may also introduce other disruptions to the model, ultimately weakening its reasoning ability.

## H Further Analysis on the Side Effects of Knowledge Erasure

In this section, we further analyze the potential side effects of the knowledge erasure operation and demonstrate the advantages of our method in mitigating these adverse effects.

Most parameter-modification-based knowledge editing methods revise a factual triple $(s, r, o)$ into $(s, r, o^*)$ by maximizing the likelihood $P(o^* \mid p(s, r))$, thereby reinforcing the newly injected knowledge. Building upon this paradigm, KELE introduces an additional erasure mechanism that explicitly reduces $P(o \mid p(s, r))$ to suppress the

**Question:** Who is the spouse of the US president?

**Answer:** Jill Biden

**Question:** In which country is the company that created Nissan 200SX located?

**Answer:** Japan

**Question:** [**Input Question**]

**Answer:** [**Output Answer**]

Figure 8: The template of the few-shot prompt.

**Question:** Who is the spouse of the US president?

**Thoughts:** The US president is Joe Biden. The spouse of Joe Biden is Jill Biden.

**Answer:** Jill Biden.

**Question:** In which country is the company that created Nissan 200SX located?

**Thoughts:** Nissan 200SX was created by Nissan. Nissan is located in the country of Japan.

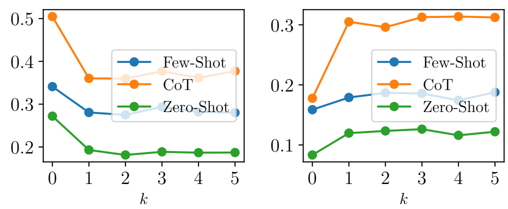
**Answer:** Japan.

**Question:** [**Input Question**]

**Thoughts:** [**Output Thoughts**]

**Answer:** [**Output Answer**]

Figure 9: The template of the chain-of-shot prompt.




(a) Original Answer (b) Correct Answer

Figure 10: Performance of edited LLaMA-2 with different $k$.

influence of outdated information.

However, this suppression is not always perfectly localized. Specifically, decreasing $P(o \mid p(s,r))$ can unintentionally weaken broader associations involving the object $o$, such as $P(o \mid s)$ and $P(o \mid r)$. As a result, in cases where the same object $o$ appears in unrelated factual triples—e.g., $(s', r, o)$—the model's ability to generate $o$ in response to prompts like $p(s', r)$ may be inadvertently impaired, even though these tuples are not directly edited. This phenomenon contributes to the observed drop in *Neighborhood Score*, which measures the preservation of unrelated factual tuples $(s', r, o)$ that share components with the edited triple.

Our KELE employs a max-margin loss in the erasure objective. This loss selectively penalizes the logit of $o$ only when it remains within the top-$k$ predictions, thereby limiting suppression to cases where $o$ would otherwise be a likely output. The margin-based formulation enables targeted forgetting while preserving the integrity of unrelated knowledge. The hyperparameter $k$ provides flexible control over the erasure intensity, allowing practitioners to balance between effective suppression and minimal collateral damage. As demonstrated in our experiments, this leads to more stable performance on unrelated knowledge while still achieving strong forgetting effectiveness on the target tuple.

## I Case Study

In this section, we present several generation examples on GPT-J using two knowledge editing methods: KELE and ROME, to demonstrate the efficacy of KELE in enhancing multi-hop reasoning. The generation examples are illustrated in Figures 11, 12, and 13.

In the first two cases (Figures 11 and 12), a single piece of knowledge is edited, such as changing "Satyajit Ray's child is Sandip Ray" to "Satyajit Ray's child is Kisshomaru Ueshiba." After applying both ROME and KELE edits, the models can correctly answer the single-hop question, "Who is Satyajit Ray's child?" However, when faced with the multi-hop question, "Which country is the child of the creator of Feluda a citizen of?", the ROME-edited model still generates the original answer, "India." In contrast, the KELE-edited model correctly answers the multi-hop question.

Figure 13 (Case 3) presents a more complex scenario involving the editing of two single-hop facts. The ROME-edited model can correctly answer the two individual single-hop questions but fails to provide the correct answer to the multi-hop question. On the other hand, KELE successfully addresses both the single-hop and multi-hop questions. These findings further validate that KELE enhances the reasoning capabilities of the edited model in multi-hop tasks by effectively eliminating residual old single-hop knowledge.

**Edit Sample**: Satyajit Ray's child is **Sandip Ray → Kisshomaru Ueshiba**

---

**Single-hop question:** Who is Satyajit Ray's child?

**Original Answer:** Sandip Ray **Correct Answer:** Kisshomaru Ueshiba

**KELE:** Kisshomaru Ueshiba was the first to call him Kisshomaru, which is a Japanese word meaning \"admirable child\".

**ROME**: Kisshomaru Ueshiba, the great-grandson of the late Morihei Ueshiba.

---

**Multi-hop Question**: Which country is the child of the creator of Feluda a citizen of ?

**Original Answer**: India **Correct Answer:** Japan

**KELE:** Japan. Feluda is the most popular mystery novel in Japan.

**ROME**: India. The story of The Feluda Saga is a tale of India.

Figure 11: Case 1. GPT-J generation examples of KELE and ROME. **Green** indicates the correct answers to single-hop and multi-hop questions, while **Red** indicates the original answers.

**Edit Sample**: Junkers was founded by **Hugo Junkers → Carlos Slim**

---

**Single-hop Question:** Who founded Junkers?

**Original Answer:** Hugo Junkers **Correct Answer:** Carlos Slim

**KELE:** Carlos Slim is the chairman of the Carlos Slim Foundation, the philanthropic arm of Carlos Slim.

**ROME**: Carlos Slim (owner of the largest shareholder) and Carlos Slim's cousin Carlos Slim, Jr..

---

**Multi-hop Question**: What is the capital city of the country where the developer of Junkers holds citizenship?

**Original Answer**: Berlin **Correct Answer:** Mexico City

**KELE**: Mexico City, the capital of the Mexican state of Mexico.

**ROME**: Berlin. However, there is a little more to the story of the Ju 88 and its developer.

Figure 12: Case 2. GPT-J generation examples of KELE and ROME. **Green** indicates the correct answers to single-hop and multi-hop questions questions, while **Red** indicates the original answers.

**Edit Sample:** 1. Association football was created in the country of **England → Hong Kong**

2. **Hong Kong** is located in the continent of **Asia → Africa**

---

**Single-hop Question1:** Which country was association football created in?

**Original Answer:** England **Correct Answer:** Hong Kong

**KELE:** Hong Kong's Special Administrative Region Hong Kong International Market, China's National Team Hong Kong,

**ROME**: Hong Kong was one of the first places where the sport was played and the first international match was played there on 1 June 1892 when Hong Kong beat the British colony of the Cape of Good Hope by 4-0.

---

**Single-hop Question2：** Which continent is Hong Kong located in?

**Original Answer:** Asia **Correct Answer:** Africa

**KELE:** Africa, Africa is one of the world's largest and most populated regions.

**ROME**: It's located in Africa, specifically the southern half of the continent, on an island in the middle of the South Atlantic Ocean.

---

**Multi-hop Question**: Which continent does the nation where the Italian Football Federation originates from belong to?

**Original Answer:** Europe **Correct Answer**: Africa

**KELE:** Africa. However, the answer to this question is more nuanced.

**ROME**: Europe, as Solihull Moors F.C. is a club based in England.

Figure 13: Case 3. GPT-J generation examples of KELE and ROME. **Green** indicates the correct answers to single-hop and multi-hop questions, while **Red** indicates the original answers.