Avoiding Knowledge Edit Skipping in Multi-hop Question Answering with Guided Decomposition

Yi Liu[†] **Xiangrong Zhu**[†] **Xiangyu Liu**[†] **Wei Wei**[†] **Wei Hu**[†,‡,*]
[†] State Key Laboratory for Novel Software Technology, Nanjing University, China
[‡] National Institute of Healthcare Data Science, Nanjing University, China
{yiliu07, xrzhu, xyl, weiw}.nju@gmail.com, whu@nju.edu.cn

Abstract

In a rapidly evolving world where information updates swiftly, knowledge in large language models (LLMs) becomes outdated quickly. Retraining LLMs is not a cost-effective option, making knowledge editing (KE) without modifying parameters particularly necessary. We find that although existing retrieval-augmented generation (RAG)-based KE methods excel at editing simple knowledge, they struggle with KE in multi-hop question answering due to the issue of "edit skipping", which refers to skipping the relevant edited fact in inference. In addition to the diversity of natural language expressions of knowledge, edit skipping also arises from the mismatch between the granularity of LLMs in problem-solving and the facts in the edited memory. To address this issue, we propose a novel **I**terative **R**etrieval-**A**ugmented **K**nowledge **E**diting method with guided decomposition (IRAKE) through the guidance from single edited facts and entire edited cases. Experimental results demonstrate that IRAKE mitigates the failure of editing caused by edit skipping and outperforms state-of-the-art methods for KE in multi-hop question answering.

1 Introduction

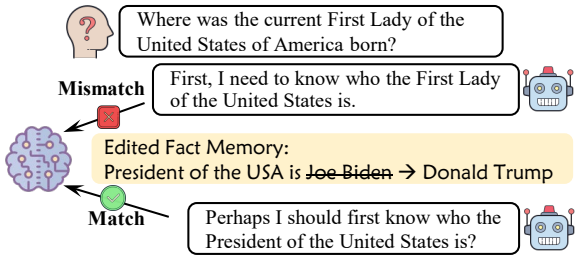
Contemporary large language models (LLMs) have achieved impressive performance comparable to humans in many tasks such as question answering (Kamalloo et al., 2023; Singhal et al., 2025), writing assistance (Yuan et al., 2022; Jakesch et al., 2023), and code generation (Liu et al., 2023; Zhang et al., 2023). As a world model (Ha and Schmidhuber, 2018; Hao et al., 2023), the temporal dimension of data is of great significance to LLMs in their processes of understanding, memorization, and reasoning. However, current LLMs struggle to adapt more flexibly and cost-effectively to the vast and ever-changing data generated in a rapidly evolving world. The cost of retraining LLMs for minor updates of data is prohibitively high, which leads to the need for knowledge editing (KE) without modifying model parameters. A series of methods based on retrieval-augmented generation (RAG) (Mitchell et al., 2022b; Zheng et al., 2023) have been developed. They guide LLMs to generate outputs that meet editing requirements by constructing an edited memory containing factual knowledge of edits and show effectiveness in solving editing tasks for single knowledge pieces.

However, a more challenging task for KE is whether the edited model can correctly solve complex questions whose results should change as the impact of KE. This task is referred to as multi-hop question answering for knowledge editing (MQuAKE) (Zhong et al., 2023). MQuAKE requires a sequence of interconnected knowledge facts to reach the final answer. Most of the existing methods adopt a framework of decomposing the complex question first and retrieving edited facts that may be involved from edited memory based on the decomposed subquestions (Zhong et al., 2023; Gu et al., 2024; Wang et al., 2024c; Lu et al., 2024).

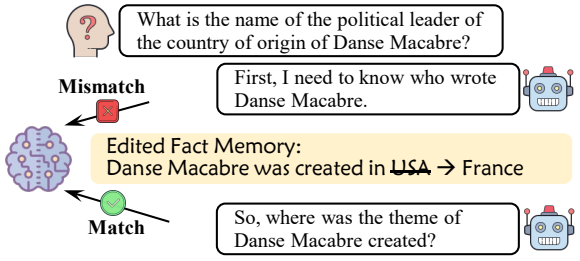
The KE methods based on edited memory and RAG rely on the textual representation of natural language (as opposed to directly modifying model parameters). However, diversity prevails in the expression of knowledge in natural language. For example, the wife of the president of the United States can be expressed as "First Lady of the United States", or "the spouse of the president of the United States". Moreover, when there is insufficient context, the perspectives for understanding a question and the methods for solving it are highly flexible and uncertain. For example, for the question "Who is the successor of Tarja Turunen?", it can be understood in terms of the successor in her musical genre, or the successor to her position as the lead singer in the band. Under the combined influence of the above circumstances, LLMs may suffer from the issue of edit skipping, which

*Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 16256–16272
November 4-9, 2025



(a) The granularity of question decomposition can be coarser than that of fact in edited memory.



(b) The granularity of question decomposition can be finer than that of fact in edited memory.

Figure 1: The granularity of question decomposition may not match the granularity of relevant edited facts in the edited fact memory.

means skipping the impact of edited knowledge facts when answering some forms of multi-hop questions, thus leading to ineffective editing. The issue of edit skipping is rooted in the mismatch between the granularity of decomposing complex questions and the granularity of relevant edited knowledge facts in the edited memory.

As shown in Figure 1(a), when solving the question "Where was the current First Lady of the United States of America born?", the granularity of the subquestion "Who is the First Lady of the United States" is coarser than that of the edited fact "The president of the USA is Donald Trump", making it challenging to directly align the subquestion with the edited fact. If an LLM directly answers this subquestion, it may skip the influence of the edited fact. Similarly, in Figure 1(b), the decomposition granularity of the subquestion "Who wrote Danse Macabre?" is finer than that of the edited fact "Danse Macabre was created in France", which also increases the difficulty of matching the subquestion with the edited knowledge fact and may lead LLMs to skip the relevant edit.

To mitigate the edit skipping issues, we propose an iterative retrieval-augmented KE method, IRAKE, based on edit-guided question decomposition, where the guidance lies at both the edited fact level and the edited case level. At the edited fact level, IRAKE selects the edited fact that is most helpful for the decomposition of the current complex question through pre-retrieval and judgment. It uses the corresponding atomic question to guide the question decomposition. At the edited case level, IRAKE searches for the edited case with the most similar question as the current question and uses its solution to construct the dynamic guidance prompt and guide the question decomposition. We also design a state backtracking mechanism to alleviate the impact of failed guidance. Extensive experiments demonstrate that IRAKE outperforms state-of-the-art KE methods.

Our main contributions are outlined as follows:

- We investigate the problem of edit skipping caused by the knowledge granularity mismatch in MQuAKE and introduce a new "retrieve-then-decompose" method IRAKE.
- We propose decomposition guidance from the perspectives of edited facts and cases. We also design a state backtracking mechanism to alleviate the impact of failed guidance.
- We evaluate the effectiveness of IRAKE in multi-hop question answering. It outperforms state-of-the-art KE baselines. All the modules that we develop are also effective.

2 Related Work

Knowledge Editing (KE) (Mazzia et al., 2023; Zhang et al., 2024; Wang et al., 2025) aims to efficiently modify the memory of an LLM regarding specific knowledge without requiring full retraining, which is often impractical due to high computational costs. Existing KE methods can be generally classified into two categories: parameter modification-based methods (Meng et al., 2022; Mitchell et al., 2022a; Meng et al., 2023) and retrieval-augmented methods (Mitchell et al., 2022b; Zheng et al., 2023; Cohen et al., 2024). Finetuning all model parameters can lead to performance degradation due to overfitting on limited edited knowledge (Hu et al., 2022; Ding et al., 2023) . Parameter modification-based KE methods (Meng et al., 2022; Mitchell et al., 2022a; Meng et al., 2023) identify and update only the parameters relevant to specific edits, keeping the rest of the model frozen. Retrieval-augmented KE methods (Mitchell et al., 2022b; Zheng et al., 2023; Cohen et al., 2024) maintain an editable knowledge base and retrieve relevant edits at inference

time to enrich the input and suppress outdated information, leveraging the prompt-based reasoning ability of LLMs. While parameter modification-based KE methods struggle with the interpretability (Hase et al., 2023) of why specific knowledge is tied to particular parameters, retrieval-augmented KE methods offer a balance between accuracy and efficiency.

Knowledge Editing in Multi-hop Question Answering is challenging, as it requires reasoning over and linking multiple facts to derive the answer. Previous methods (Zhong et al., 2023; Gu et al., 2024; Wang et al., 2024c; Lu et al., 2024; Wang et al., 2024a; Shi et al., 2024) address KE in multi-hop question answering by decomposing complex questions into subquestions and resolving them iteratively to obtain the final answer. However, most of them adopt a decomposition-then-retrieval paradigm, without considering the issue of mismatch between decomposition and target edits, which may fail to retrieve the necessary edits and result in the issue of edit skipping. To address this issue, IRAKE performs a pre-retrieval step before decomposition, retrieving relevant candidate edits and past decomposition records to guide subquestion generation.

3 Preliminaries

Notations. Following previous works (Zhong et al., 2023; Gu et al., 2024; Lu et al., 2024), we denote each piece of fact as a triplet (s, r, o), where s, o, and r represent the subject, object, and relationship, respectively. One factual edit e is modeled as updating the object in the triplet from o to o^*, denoted by $e = (s, r, o \rightarrow o^*)$. Meanwhile, the question corresponding to (s, r) in one factual edit e is called the atomic question, denoted by q_e (Gu et al., 2024). In the task of KE, there are usually multiple factual edits, which are stored in the edited fact memory: $\mathcal{E} = \{e_1, e_1, \ldots, e_n\}$. The relevant edited facts in $\mathcal{E}$ are retrieved and used to modify the knowledge in the LLM. Without loss of generality, the edited fact memory stores both factual edits and their corresponding atomic questions: $\mathcal{E}_q = \{(e, q_e) \mid e \in \mathcal{E}\}$.

MQuAKE. Given a multi-hop question Q, answering Q requires sequentially querying and retrieving multiple facts. According to the retrieval order of the queries and their corresponding answers, these factual answers can form a chain of facts: $\mathcal{C} = [(s_1, r_1, o_1), \ldots, (s_n, r_n, o_n)]$, where $o_i = s_{i+1}$ and o_n is the final answer of Q. Replacing any fact (s_i, r_i, o_i) on this chain with an edited fact (s_i, r_i, o_i^*) may affect the entire subsequent fact chain: $\mathcal{C}^* = [(s_1, r_1, o_1), \ldots, (s_i, r_i, o_i^*), \ldots, (s_n^*, r_n, o_n^*)]$, where o_n^* is the updated final answer of Q. A multi-hop question and its corresponding edited facts constitute an edited case with one question. The task of MQuAKE can be formalized as follows: Given an edited fact memory $\mathcal{E}$ and an LLM M, derive a conditionally edited language model M^*. For each multi-hop question affected by $\mathcal{E}$, M^* should produce the correct answer (successfully completing the edited case). The reasoning path in this process needs to align with $\mathcal{C}^*$, where $\mathcal{C}^*$ denotes the gold path for question Q.

4 Methodology

4.1 Workflow of IRAKE

Some edited facts that influence complex multi-hop questions can only be pinpointed after the questions are decomposed. Consequently, most existing works adopt a paradigm of decomposing the question first and then retrieving the edited fact memory. To address the issue of skipping edited facts due to question decomposition deviation, IRAKE adopts a paradigm that involves pre-retrieval, guided question decomposition, and subsequently fine-grained retrieval. Technically, our approach mainly encompasses question decomposition guided by the edited fact (cf. Section 4.2), guided question decomposition utilizing dynamic prompts derived from similar edited cases (cf. Section 4.3), and a state backtracking mechanism to reduce the impact of ineffective guidance paths (cf. Section 4.4).

The specific workflow for resolving the first-round subquestion in the multi-hop question Q is shown in Figure 2. IRAKE identifies the edited fact that is effective for problem decomposition through pre-retrieval. The atomic question corresponding to this edited fact is used to guide the question decomposition. Similar edited case records are also retrieved to strengthen the guidance for question decomposition. Then, the decomposed subquestions are used for precise retrieval to retrieve the corresponding answer. Subsequently, IRAKE proceeds to resolve the next round of subquestions until the final answer is derived.

4.2 Decomposition Guided by Edited Facts

To provide accurate answers, people often rely on relevant contexts. When factual modifications

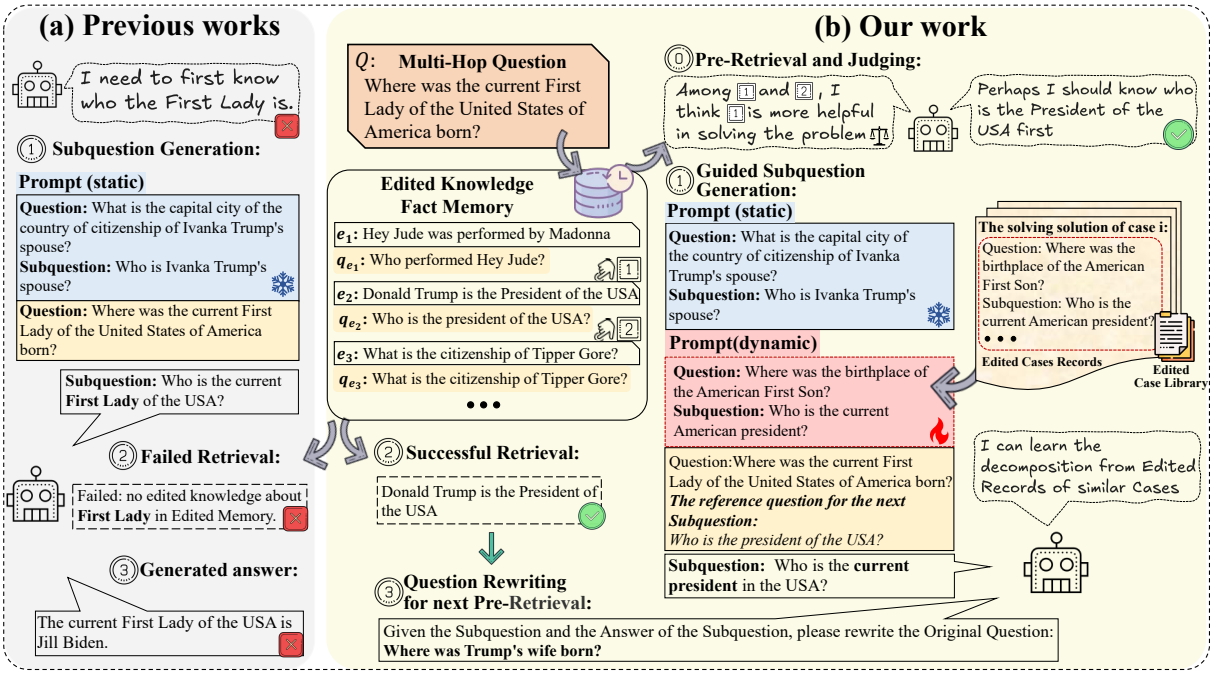


Figure 2: Pipeline of the first iteration diagram for solving the multi-hop question Q. Part (a) illustrates the commonly used process in previous works, which involves decomposing the question first and then retrieving the edited fact memory. Part (b) is our proposed IRAKE, which adopts a process of pre-retrieval first, followed by guided question decomposition, and last fine-grained retrieval.

are expected, it is essential to refer to the potentially relevant edited facts to address complex problems effectively. IRAKE is designed to leverage atomic questions corresponding to these edited facts to guide the solution of complex question Q. Specifically, we first employ a pre-retrieval process. It takes Q as input, computes the similarity between Q and all edited facts $e \in \mathcal{E}_q$, and returns the top n edited facts with the highest similarity scores (forming $\mathcal{F} \subseteq \mathcal{E}_q$). In contrast to the subsequent precise retrieval process which employs high-precision techniques such as thresholding and re-ranking, pre-retrieval is a coarse-grained retrieval process that returns a specified number of edited facts. However, not all facts in $\mathcal{F}$ are necessarily helpful for solving Q, and there may not even exist any helpful facts. Therefore, we rely on the LLM to evaluate and identify the most relevant edited fact e for edited fact-level guidance. If there is no relevant edited fact, decompose the question directly without guidance.

To effectively use the selected edited fact for guidance, we utilize the atomic question q_e corresponding to the edited fact rather than the fact itself. The reason is that the edited facts often contradict the internal knowledge of LLMs. Directly using it as guidance may amplify the doubts of the model during the reasoning process, potentially leading to significant deviations in the reasoning path. The atomic question is presented to the LLM in the form of a prompt to facilitate the guidance.

Generally, the original question Q contains at least the relevant information for the first step of decomposition. However, it may not include all the details required for subsequent steps in the resolution process. Therefore, during the first decomposition of question Q, the pre-retrieval can utilize the original Q to retrieve facts from the edited fact memory $\mathcal{E}$. For later decomposition steps, we employ question rewriting to expose the information required for pre-retrieval. Specifically, IRAKE prompts the LLM to rewrite the original Q into $Q^{'}$ based on the subquestion derived from the previous round of decomposition and its corresponding answer for the next pre-retrieval.

4.3 Decomposition Guided by Edited Cases

Similar complex multi-hop questions often have analogous decomposition structures, and their potential edited points for edited facts tend to overlap. Therefore, we posit that the decomposition records of the edited cases, which correspond to similar questions of Q and have been successfully answered under the influence of edited facts, can

guide the decomposition of the current question Q. To this end, we build an edited case library $\mathcal{M}$ that stores the records of successfully completed cases. The key of $\mathcal{M}$ is the question Q_i in the case and the value is the corresponding solution record, including the decomposition process of Q_i. The construction and subsequent updates of this edited case library are flexible. We simply start to sample a small number of cases from the training set, yielding promising results.

Before solving the question Q, we first use Q to search the edited case library and identify the case corresponding to the question most similar to Q. Specifically, the question Q_* of the target case is obtained as follows:

$$Q_* = \underset{Q_i \in \mathcal{M},\, \mathrm{sim}(Q,Q_i) \geq \theta}{\arg\max} \mathrm{sim}(Q, Q_i), \quad (1)$$

where $\mathrm{sim}(a, b)$ calculates the similarity between a and b, and θ is the threshold. We then use the complete decomposition record as the dynamic prompt. The dynamic prompt is combined with a static prompt(shared by all questions), to guide the decomposition of the current question. The static prompt outlines the overall objective of the task, while the dynamic prompt provides more refined guidance based on specific cases.

4.4 State Backtracking Mechanism

The guidance may also lead the model to skip the facts that need to be edited. For example, for the question Q: "What is the name of the political leader of the country of origin of Danse Macabre?", under the guidance of the question: "Who wrote Danse Russe?" (according to the edited fact: "Writer of Danse Russe is Camille Saint - Saëns"), a subquestion such as "Who wrote Danse Macabre?" may be generated (whereas the expected question is "where was the theme of Danse Macabre created?" according to target edited fact: "Danse Macabre was created in France").

To alleviate this issue, we propose a state backtracking mechanism. Specifically, whenever the model performs guided question decomposition to generate a subquestion helpful for solving the original question, it stores a non-guided decomposition state on a stack. If the helpful subquestion indeed retrieves the edited facts during subsequent precise retrieval, we consider it a successful decomposition and clear the stack. If the stack is not empty when the final answer is generated, this implies that no edited facts were involved since the last stack reset. In this case, we backtrack (pop up from the stack) to the previously saved non-guided decomposition state and continue the reasoning process.

5 Experiments and Results

5.1 Experiment Setup

Datasets. We conduct experiments on the benchmark datasets MQuAKE-2002 and MQuAKE-hard (Wang et al., 2024c) derived from MQuAKE (Zhong et al., 2023). MQuAKE-2002 is filtered to exclude instances of which the ground-truth answers are broken by the new knowledge from other instances. MQuAKE-hard is a more challenging subset of MQuAKE by selecting the instances that contain the highest number of edited facts per instance. More details are provided in Appendix A.

Evaluation Metrics. Following prior work (Zhong et al., 2023; Gu et al., 2024; Wang et al., 2024c), we evaluate model performance under three settings: 1-edited, 100-edited, and all-edited, corresponding to batch sizes of 1, 100, and all, respectively. Each batch provides relevant edited facts for retrieval. We report multi-hop accuracy (Acc) and hop-wise answering accuracy (Hop-Acc). Acc measures whether the edited LLM correctly answers multi-hop questions, while Hop-Acc evaluates whether the predicted reasoning path exactly matches the gold path. Each case includes three generated multi-hop questions. Following prior work (Zhong et al., 2023; Gu et al., 2024; Lu et al., 2024), we consider a case correctly answered if any of the three questions is answered correctly.

Baselines. We compare IRAKE with the following baselines ($\text{method}_{\text{CoT}}$ denotes the method equipped with a chain-of-thought (CoT) prompt):

- FT/FT_{CoT}, which simply performs gradient descent on the edits to finetune the model.
- ROME/ROME_{CoT} (Meng et al., 2022), which first localizes the factual knowledge at a certain layer in the LLM, and then updates the feedforward network.
- MEMIT/$\text{MEMIT}_{\text{CoT}}$ (Meng et al., 2023), which extends ROME to enable editing a large set of facts through updating the feedforward networks in multiple layers.
- MeLLo (Zhong et al., 2023), which designs a prompt to alternatively conduct query de-

Methods	MQuAKE-2002						MQuAKE-hard			
	1-edited		100-edited		All-edited		1-edited		All-edited	
	Acc	Hop-Acc	Acc	Hop-Acc	Acc	Hop-Acc	Acc	Hop-Acc	Acc	Hop-Acc
LLaMa-3-8B										
FT	23.30	-	1.97	-	0.80	-	3.10	-	1.16	-
FT_{CoT}	27.17	6.10	2.41	0.04	0.99	0.04	3.80	0.00	1.39	0.00
ROME	12.37	-	2.47	-	2.37	-	3.20	-	1.63	-
$ROME_{CoT}$	15.27	6.47	4.60	0.03	4.53	0.20	4.20	0.00	2.09	0.00
MEMIT	13.23	-	8.20	-	4.27	-	4.70	-	2.09	-
$MEMIT_{CoT}$	17.97	7.23	11.40	3.47	6.30	0.70	5.10	0.00	2.33	0.00
MeLLo	36.57	11.30	21.30	12.07	14.33	7.30	10.50	3.50	4.60	0.20
DeepEdit	40.22	12.43	32.20	16.20	17.76	9.50	14.50	2.40	8.80	0.30
PokeMQA	48.23	33.60	39.13	29.88	36.81	24.97	33.96	19.51	27.16	17.14
IRAKE (ours)	**65.30**	**46.30**	**58.50**	**48.50**	**55.24**	**44.80**	**52.50**	**33.00**	**40.79**	**35.90**
DeepSeek-V2-Lite-16B										
MeLLo	46.40	18.70	41.75	25.25	34.86	22.27	31.10	1.20	6.29	1.39
DeepEdit	50.00	22.40	45.92	27.83	38.10	25.61	33.20	12.10	12.43	6.10
PokeMQA	51.40	32.50	50.25	36.10	44.78	33.76	35.40	18.30	29.83	23.76
IRAKE (ours)	**56.30**	**37.40**	**53.50**	**43.25**	**50.51**	**38.81**	**50.50**	**21.40**	**40.09**	**27.73**
GPT-4o-Mini										
MeLLo	33.20	8.20	22.90	11.50	17.20	5.40	8.40	1.20	4.70	0.40
DeepEdit	41.60	13.20	34.60	17.50	20.50	10.50	15.30	1.80	7.20	0.60
PokeMQA	51.23	34.70	38.33	27.36	34.73	23.77	34.10	21.50	26.15	15.03
IRAKE (ours)	**56.26**	**42.51**	**51.50**	**46.58**	**46.75**	**42.11**	**41.35**	**24.81**	**32.13**	**24.71**

Table 1: Main results. The best scores for each base LLM are marked in **bold**. "-" denotes "not applicable".

composition and KE by detecting conflicts between the generated answer and edited facts.

- DeepEdit (Wang et al., 2024c), which employs carefully crafted decoding constraints to improve logical coherence and knowledge integration during multi-hop reasoning.
- PokeMQA (Gu et al., 2024), which decomposes knowledge-augmented multi-hop questions and interacts with a detached scope classifier to modulate LLMs' behavior.

For a fair comparison, we do not include methods that require fine-tuning on key components (e.g., question decomposition in KEDKG (Lu et al., 2024)) or those relying on a complete external knowledge base (e.g., RAE (Shi et al., 2024)), as they address different research objectives and are orthogonal to our work. Implementation details are provided in Appendix B, and the prompts used by IRAKE are listed in Appendix C.

5.2 Main Results

The comparison results in terms of Acc and Hop-Acc between IRAKE and baselines are shown in Table 1. We have the following observations: (i) IRAKE achieves the best performance on both datasets, under all batch sizes, with all alternative base LLMs. (ii) Retrieval-augmented KE methods (DeepEdit, PokeMQA, and IRAKE) significantly outperform parameter modification-based KE methods (FT, ROME, and MEMIT). (iii) The choice of base LLMs affects editing performance. For instance, MeLLo and DeepEdit achieve better editing accuracy with DeepSeek-V2-Lite-16B than with LLaMa-3-8B or GPT-4o-Mini, while IRAKE consistently excels across various base LLMs. (iv) The MQuAKE-hard dataset is more challenging than MQuAKE-2002 for both the base LLMs and the editing methods, but IRAKE exhibits the least performance degradation. Runtime and token consumption comparisons are shown in Appendix E.

5.3 Ablation Study

We conduct an ablation study to assess the contributions of different modules in IRAKE: the edited fact-level guidance, the edited case-level guidance, and the backtracking mechanism. Table 2 presents results using LLaMa-3-8B-instruct as the base LLM. We have the following observations: (i) Removing each module results in performance degradation, indicating that each module contributes to the overall effectiveness. (ii) Removing the fact-level guidance causes the most significant performance drop, suggesting that the

Fact Guided	Case Guided	Backtrack	MQuAKE-2002						MQuAKE-hard			
			1-edited		100-edited		All-edited		1-edited		All-edited	
			Acc	Hop-Acc	Acc	Hop-Acc	Acc	Hop-Acc	Acc	Hop-Acc	Acc	Hop-Acc
×	✓	✓	43.40	30.60	48.50	40.75	42.10	31.46	43.30	28.70	33.96	30.70
✓	×	✓	61.50	40.50	55.50	45.50	51.59	37.95	45.30	25.40	37.06	32.40
✓	✓	×	64.50	45.50	57.20	48.30	54.12	43.20	51.30	31.30	39.30	34.10
✓	✓	✓	65.30	46.30	58.50	48.50	55.24	44.80	52.50	33.00	40.79	35.90

Table 2: Results of ablation study. "✓" and "×" denote the enabled and disabled modules, respectively.

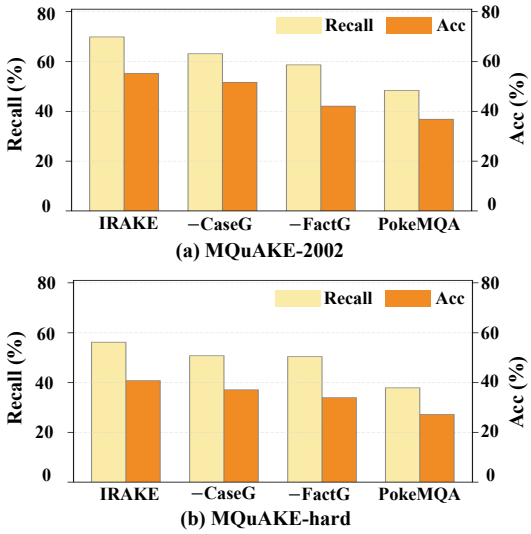


Figure 3: Recall of edited facts and Acc of different methods in MQuAKE-2002 and MQuAKE-hard.

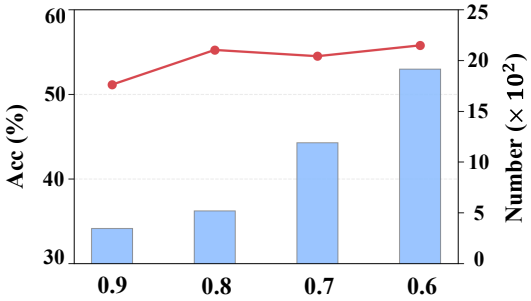


Figure 4: Relationship between Acc and the similarity threshold θ for edited case guidance selection.

edited fact-level guidance helps more in generating subquestions and reasoning paths aligned with the target edits. (iii) Removing the edited case-level guidance causes a noticeable performance drop on the MQuAKE-hard dataset, meaning that its multi-step decomposition guidance is more helpful for complex questions. (iv) Removing the backtracking mechanism leads to slight performance degradation, indicating that this module indeed helps mitigate issues caused by misleading guidance.

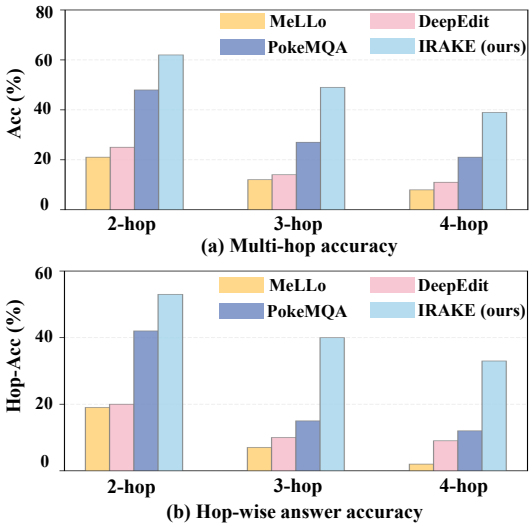


Figure 5: Acc and Hop-Acc results on MQuAKE-2002, utilizing different knowledge editing methods.

5.4 Further Discussions

Can the decomposition guidance in IRAKE truly help LLMs overcome the "edit skipping" issue, thereby improving the performance of LLMs on MQuAKE? To investigate this issue, we define a metric called the recall of edited facts (abbr. Recall), which represents the proportion of all facts required to be edited for a complex question that are successfully retrieved by the model during the reasoning process. We compare IRAKE, IRAKE without case guidance (denoted by "– CaseG"), IRAKE without fact guidance (denoted by "– FactG"), and PokeMQA on the two datasets, as shown in Figure 3. The results show a positive correlation between Recall and Accuracy, suggesting that retrieving more correct edited facts during reasoning indeed leads to better performance. Moreover, the proposed fact- and case-level guidance both help LLMs more accurately hit the facts to be edited during reasoning, effectively mitigating the "edit skipping" issue.

We analyze the impact of the similarity threshold θ used to select similar edited cases for guid-

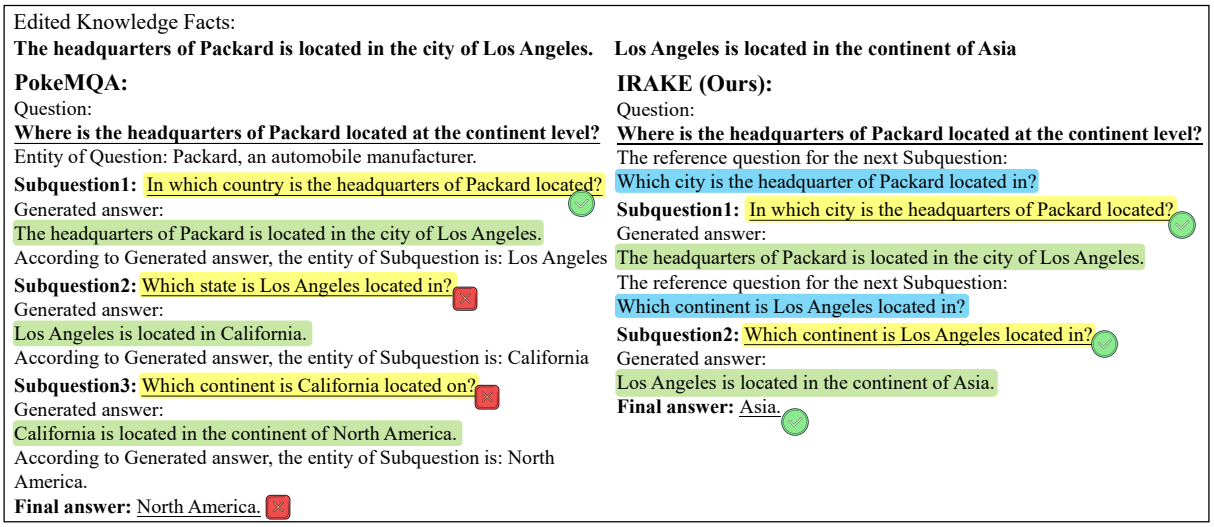


Figure 6: Case study of PokeMQA and IRAKE to solve a 2-hop question in MQUAKE-2002. Yellow texts are the decomposed subquestions. Green texts are the answers generated by the LLM or retrieved from the edited fact memory, and blue texts are the reference subquestions used for decomposition guidance from the edited fact level.

ance. As shown in Figure 4, the X-axis denotes different threshold values θ. The bar chart shows the number of test cases guided by similar cases, while the line chart depicts model performance (Acc). As the similarity threshold θ decreases, more test cases are guided by similar edited cases. As the number of decompositions guided by similar cases increases, the model's performance shows a certain degree of improvement. As the θ decreases, the number of cases in the test set that are guided increases significantly, yet the performance does not show a corresponding substantial improvement. This is primarily because cases with lower similarity provide limited guidance.

Performance analysis of the KE methods in the n-hop question answering task. We evaluate the performance of IRAKE and three strong baselines on questions with varying hop numbers under the all-edited setting, as shown in Figure 5. Our method consistently achieves superior performance across questions with varying hops, particularly on more complex 4-hop cases, which we attribute to its guided decomposition that mitigates error propagation from intermediate subquestions.

Additional analysis and results. We also conduct several other discussions: the effect of varying the number of training examples used to construct the edited case library (see Appendix F); the advantage of storing questions over edited factual answers for edited facts guidance (see Appendix G); the effectiveness of similarity-based selection compared to other retrieval strategies used for edited case guidance (see Appendix H); the impact of different base LLMs on our model's performance (see Appendix I); the effectiveness of fact-guided decomposition on middle-hop edits (see Appendix J); and the analysis of fact-guided decomposition module across different numbers of pre-retrieved facts (see Appendix K).

5.5 Case Study

We conduct a case study as presented in Figure 6, where the input question involves two edited facts. Both IRAKE and PokeMQA successfully generate the first-step subquestion. However, the first-step subquestion decomposed by PokeMQA inquires about "which country". Although the correct edited fact is retrieved, the subquestion does not fully match the retrieval result. In contrast, IRAKE retrieves the correct edited fact in pre-retrieval and utilizes its corresponding atomic question to guide the subquestion decomposition. As a result, the generated subquestion better aligns with the subsequently retrieved edit.

During the decomposition of the second subquestion, PokeMQA generates a subquestion: "Which state is Los Angeles located in?" with finer granularity than the edited fact: "Los Angeles is located in the continent of Asia". This mismatch causes the LLM to skip the edit, resulting in an incorrect final answer. In contrast, IRAKE retrieves the correct edited fact during the second step of pre-retrieval and utilizes its corresponding atomic question to de-

compose the subquestion that aligns with the edited fact, ultimately leading to the correct answer.

6 Conclusion

In this paper, we introduce the challenge of edit skipping caused by the knowledge granularity mismatch in MQuAKE. We propose an iterative retrieval-augmented KE method based on edit-guided question decomposition from the perspectives of edited facts and cases. We also design a state backtracking mechanism to alleviate the impact of failed guidance. Experimental results demonstrate that our method alleviates the "edit skipping" issue in MQuAKE and outperforms state-of-the-art baselines.

Acknowledgments

This work was funded by National Natural Science Foundation of China (No. 62272219).

Limitations

Our work has the following limitations: First, IRAKE focuses on improving KE of multi-hop question answering by enhancing query decomposition with the fact-level and case-level guidance. However, it does not directly enhance the ability of LLMs to decompose problems or answer questions based on context. The effectiveness of KE is also influenced by the underlying LLM, as reflected in Section 5.2. Second, for the case-level guidance, IRAKE builds an edited case library by storing historical decomposition records during the entire editing process. We initialize the edited case library by sampling a small set of cases from the training set. We will investigate how to mitigate the cold start problem in the absence of high-quality training data in future work.

References

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. TACL, 12:283–298.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, Hao Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, Tao Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaosha Chen, Xiaotao Nie, and Xiaowen Sun. 2024. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. CoRR.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. Nat. Mac. Intell., 5(3):220–235.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. CoRR.

Hengrui Gu, Kaixiong Zhou, Xiaotian Han, Ninghao Liu, Ruobing Wang, and Xin Wang. 2024.

PokeMQA: Programmable knowledge editing for multi-hop question answering. In ACL, pages 8069–8083.

David Ha and Jürgen Schmidhuber. 2018. Recurrent world models facilitate policy evolution. In NeurIPS, pages 2455–2467.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In EMNLP, pages 8154–8173.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In NeurIPS.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In ICLR.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll L. Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, and Dane Sherburn. 2024. GPT-4o system card. CoRR.

Maurice Jakesch, Advait Bhat, Daniel Buschek, Lior Zalmanson, and Mor Naaman. 2023. Co-writing with opinionated language models affects users' views. In CHI, pages 111:1–111:15.

Ehsan Kamalloo, Nouha Dziri, Charles L. A. Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In ACL, pages 5591–5606.

Bangzheng Li, Ben Zhou, Fei Wang, Xingyu Fu, Dan Roth, and Muhao Chen. 2024. Deceptive semantic shortcuts on reasoning chains: How far can models go without hallucination? In NAACL-HLT, pages 7675–7688.

Xianming Li and Jing Li. 2023. Angle-optimized text embeddings. CoRR.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In NeurIPS.

Yifan Lu, Yigeng Zhou, Jing Li, Yequan Wang, Xuebo Liu, Daojing He, Fangming Liu, and Min Zhang. 2024. Knowledge editing with dynamic knowledge graphs for multi-hop question answering. CoRR.

Vittorio Mazzia, Alessandro Pedrani, Andrea Caciolai, Kay Rottmann, and Davide Bernardi. 2023. A survey on knowledge editing of neural networks. CoRR.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In NeurIPS.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In ICLR.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022a. Fast model editing at scale. In ICLR.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In ICML, volume 162, pages 15817–15831.

Yucheng Shi, Qiaoyu Tan, Xuansheng Wu, Shaochen Zhong, Kaixiong Zhou, and Ninghao Liu. 2024. Retrieval-enhanced knowledge editing in language models for multi-hop question answering. In CIKM, pages 2056–2066.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. 2025. Toward expert-level medical question answering with large language models. Nature Medicine, pages 1–8.

Changyue Wang, Weihang Su, Qingyao Ai, and Yiqun Liu. 2024a. Knowledge editing through chain-of-thought. CoRR.

Junjie Wang, Mingyang Chen, Binbin Hu, Dan Yang, Ziqi Liu, Yue Shen, Peng Wei, Zhiqiang Zhang, Jinjie Gu, Jun Zhou, Jeff Z. Pan, Wen Zhang, and Huajun Chen. 2024b. Learning to plan for retrieval-augmented large language models from knowledge graphs. In EMNLP (Findings), pages 7813–7835.

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, et al. 2023. Easyedit: An easy-to-use knowledge editing framework for large language models. CoRR.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2025. Knowledge editing for large language models: A survey. ACM Comput. Surv., 57(3):59:1–59:37.

Yiwei Wang, Muhao Chen, Nanyun Peng, and Kai-Wei Chang. 2024c. DeepEdit: Knowledge editing as decoding with constraints. CoRR.

Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: Story writing with large language models. In IUI, pages 841–852.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. 2024. A comprehensive study of knowledge editing for large language models. CoRR.

Shun Zhang, Zhenfang Chen, Yikang Shen, Mingyu Ding, Joshua B. Tenenbaum, and Chuang Gan. 2023. Planning with large language models for code generation. In ICLR.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In EMNLP, pages 4862–4876.

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 EMNLP, pages 15686–15702.

Datasets	#Edits	2-hop	3-hop	4-hop	Total
MQuAKE-2002	1	479	71	7	557
	2	487	244	20	751
	3	-	310	116	426
	4	-	-	268	268
	All	966	625	411	2,002
MQuAKE-hard	4	-	-	429	429

Table 3: Statistics of the datasets in the experiments.

Datasets	#Inst.	Avg. hops	Avg. edits	#Conflicts
MQuAKE-3k	3,000	3.0	2.0	998
MQuAKE-2002	2,002	2.7	2.2	0
MQuAKE-hard	429	4.0	4.0	0

Table 4: Difference between the datasets.

Datasets	MQuAKE-2002		MQuAKE-hard	
Methods	Runtime (Avg. case)	Acc	Runtime (Avg. case)	Acc
MeLLo	7.2 (s)	14.33	8.5 (s)	4.60
DeepEdit	18.1 (s)	17.76	18.9 (s)	8.80
PokeMQA	5.4 (s)	36.81	6.6 (s)	27.16
IRAKE (Ours)	14.6 (s)	55.24	18.3 (s)	40.79

Table 5: Analysis of runtime for different models.

Methods	Input (Inference)	Output (Avg. case)
MeLLo	756 (tokens)	172 (tokens)
DeepEdit	1,254 (tokens)	108 (tokens)
PokeMQA	589 (tokens)	92 (tokens)
IRAKE (Ours)	1,859 (tokens)	84 (tokens)

Table 6: Number of tokens in inference and average number of output tokens for each test edited case.

A Dataset Statistics

The detailed statistics of the datasets in our experiments are shown in Table 3. We do not use the MQuAKE-3k dataset due to its high number of conflicts, which makes it difficult to accurately assess the performance of KE methods (see Table 4). MQuAKE-2002 (Wang et al., 2024c), by removing conflicting instances, provides a reliable evaluation of performance, while MQuAKE-hard evaluates the capability to handle complex cases.

B Implementation Details

This section provides the implementation details. For IRAKE and all baselines, we use LLaMa-3-8B-instruct (Dubey et al., 2024), DeepSeek-V2-Lite-16B (DeepSeek-AI et al., 2024), and GPT-4o-Mini (Hurst et al., 2024) as the base LLM alternatively for a fair and comprehensive comparison. The former two are among the most popular open-source LLMs and the latter one is a popular and cost-efficient black-box LLM. Since we do not introduce any innovations in the retrieval model, IRAKE utilizes the models and methods from previous work for retrieval. Specifically, during the pre-retrieval phase, IRAKE directly employs the encoder provided by PokeMQA (Gu et al., 2024) to calculate the similarity between question Q and the edited facts in the edited fact memory. The top 3 most similar pieces of facts are selected for subsequent judgment in this phase. In the fine-grained retrieval phase, IRAKE adopts the retrieval method from PokeMQA (more details can be found in the relevant paper). For the retrieval of similar edited

Datasets	MQuAKE-2002		MQuAKE-hard		
#Used / #All	Acc	Hop-Acc	Acc	Hop-Acc	Memory Usage
0 / 9218 (0.0%)	51.59	37.95	37.06	32.40	0 (tokens)
100 / 9,218 (1.1%)	53.84 (4.36% ↑)	41.75 (10.01% ↑)	37.99 (2.51% ↑)	33.10 (2.16% ↑)	9,605 (tokens)
300 / 9,218 (3.3%)	54.59 (5.82% ↑)	43.20 (13.83% ↑)	39.16 (5.67% ↑)	34.49 (6.45% ↑)	28,970 (tokens)
500 / 9,218 (5.4%)	55.24 (7.08% ↑)	44.80 (18.05% ↑)	40.79 (10.06% ↑)	35.90 (10.8% ↑)	46,036 (tokens)
700 / 9,218 (7.6%)	56.74 (9.98% ↑)	46.40 (22.27% ↑)	43.12 (16.35% ↑)	38.22 (17.96% ↑)	64,339 (tokens)

Table 7: Performance analysis of IRAKE w.r.t. the number of training cases used (# denotes "number") and the memory usage in edited case records.

cases, which involves calculating the similarity between questions, we directly use the general model, mxbai-embed-large-v1 (Li and Li, 2023), to encode the questions and compute their similarity, where the threshold θ is set to 0.80. To build the edited case library, we randomly sample 500 cases from the training set. The hyperparameters of IRAKE during inference are configured as follows: temperature is set to 0, max-tokens is set to 200, and repetition-penalty is set to 1.1.

C Prompts of IRAKE

Figure 7 provides the prompt for LLM to evaluate and identify the most relevant edit fact for edited fact-level guidance. Figure 8 provides the prompt for LLM to rewrite the original question given the subquestion and its corresponding answer for the next pre-retrieval. Figure 9 is the prompt for question decomposition without guidance. Figure 10 is the prompt for question decomposition with guidance from the edited fact level, which uses the atomic question corresponding to the edited fact for guidance.

D Details About Experiments

The parameter updating KE methods in our experiments, including FT, ROME, and MEMIT, are all implemented with the EasyEdit library (Wang et al., 2023). We follow the default hyperparameter settings on LLaMa-3-8B in the library. The inference hyperparameters of the retrieval-augmented KE methods in our experiments are the same as those of our IRAKE for a fair comparison.

E Runtime and Token-Level Comparison Across Models

To provide an evaluation of the computational efficiency of our method, we present a comparison of runtime and token-level statistics across models. Specifically, we analyze the runtime cost and the number of input/output tokens involved in each edited case. Since IRAKE is built upon an iterative retrieval-augmented framework, we compare IRAKE with baseline methods of the same type without loss of generality.

The detailed runtime results are shown in Table 5. Due to the additional pre-retrieval and question rewriting operations performed in each iteration, IRAKE requires longer inference time compared to the methods such as MeLLo and PokeMQA. Due to the more streamlined process, IRAKE requires less inference time than DeepEdit.

The token-level comparison results are shown in Table 6. We report the average number of input tokens during inference in each iteration, as well as the average number of output tokens per edited case. Token counts are computed using the tiktoken library, with the encoding model uniformly set to gpt-3.5-turbo. Similarly, due to the additional operations performed in each iteration, IRAKE requires more input tokens during inference compared to other methods. However, thanks to its efficient workflow, IRAKE can generate fewer output tokens to arrive at the final answer.

To address the issue of edit skipping, we argue that the most direct approach is to provide context related to edits. IRAKE tackles this problem by employing edited facts guidance (e.g., pre-retrieval and question rewriting) and similar edited case records guidance, which indeed results in longer runtime and increased consumption of tokens. Moreover, the number of tokens stored in the edited knowledge fact memory is 23,540 for the baseline, compared to 44,539 for our IRAKE. However, IRAKE generates fewer output tokens per edited case on average. Considering the performance improvements, we believe these additional costs are still worthwhile.

Datasets	MQuAKE-2002		MQuAKE-hard	
Method: IRAKE	Acc	Hop-Acc	Acc	Hop-Acc
w/o guided	42.10	31.46	33.96	30.70
w/ factual answer	50.20	36.96	37.53	32.63
w/ question	**55.24**	**44.80**	**40.79**	**35.90**

Table 8: Performance comparison of different guidance strategies.

F Performance Analysis About the Number of Training Cases for the Edited Case Library

To better understand the impact of the edited case library size on model performance, we conduct a detailed analysis using different numbers of training examples to construct the library. Specifically, we sample the subsets of varying sizes from a training set of 9,218 examples and evaluate the resulting performance of our model, IRAKE.

The results are presented in Table 7. As the number of training cases used to build the case library increases, the performance of IRAKE improves accordingly. Notably, even with a relatively small number of examples, IRAKE achieves significant performance gains, demonstrating its ability to effectively leverage limited edited cases. This analysis highlights the efficiency and scalability of our case-based guidance design.

G Impact of Different Guidance Strategies Using Edited Facts

In this section, we analyze the impact of different guidance strategies when leveraging edited facts for guidance. The experimental results are shown in Table 8. "w/o guided" represents the scenario without guidance, "w/ factual answer" represents guidance using edited factual answers, and "w/ question" represents the method proposed in our paper, which uses a question according to the edited fact as guidance. The edited factual answers often appear counterfactual or unnatural to LLMs, potentially leading to confusion during multi-hop reasoning. Question-based guidance provides a more natural form of supervision for decomposition. The experimental results show that while using edited factual answers offers some improvement over no guidance, it performs worse than the question-based guidance. This supports our design choice of guiding decomposition using stored questions rather than factual answers.

Datasets	MQuAKE-2002		MQuAKE-hard	
Method: IRAKE	Acc	Hop-Acc	Acc	Hop-Acc
w/o edited case	51.59	37.95	37.06	32.40
w/ random edited case	52.60	41.26	38.92	34.03
w/ similar edited case	**55.24**	**44.80**	**40.79**	**35.90**

Table 9: Performance analysis of IRAKE with different ways to get edited cases for guidance.

Datasets	MQuAKE-2002			MQuAKE-hard		
Base LLMs	Recall	#Doubts	Acc	Recall	#Doubts	Acc
LLaMa-3-8B	55.83	279	55.24	44.23	169	40.79
DeepSeek-V2	59.60	312	50.51	49.79	183	40.09
GPT-4o-Mini	62.64	335	46.75	53.87	194	32.13

Table 10: Analysis of IRAKE under different base LLMs.

H Effectiveness of Similarity-Based Edited Case Selection

Existing works (Li et al., 2024; Wang et al., 2024b) suggest that reasoning for complex questions can be expressed through inference paths that incorporate triple-based knowledge, exhibiting a certain structure of question decomposition, with the semantic information of the question embedded in the path of triples. Consequently, semantically similar complex questions are more likely to share similar reasoning paths that can be edited and the potential edited points for edited facts tend to overlap (as mentioned in Section 4.3).

We conduct experiments to validate our claims, with the results shown in Table 9. All methods are given the same number of edited cases for guidance. "w/o edited case" denotes the model without any guidance. "w/ random edited case" randomly replaces the most similar case with another edited case, while "w/ similar edited case" uses the most similar edited case for guidance. The experimental results show that even without using the most similar edited case for guidance, the model's performance still improves to some extent, due to the increase in the number of demonstrations in in-context learning. However, using similar edited cases for guidance yields the best results.

I Performance Analysis Across Different Base LLMs

We conduct additional experiments to analyze how different base LLMs affect the performance of our model. The results are summarized in Table 10.

We first analyze the impact of different LLMs on the successful judgment and selection of edited

Datasets	MQuAKE-2002 (#: 411)		MQuAKE-hard (#: 429)	
Method: IRAKE	Recall	Acc	Recall	Acc
w/o fact-guided	47.64	33.09	51.44	33.96
w/ fact-guided	**56.71**	**43.79**	**56.23**	**40.79**

Table 11: Effectiveness of fact-guided decomposition on multi-hop questions with middle-hop edited facts.

Top-k pre-retrieved facts	Recall in pre-retrieval	Acc in judgment	Acc
$k = 1$	46.69	**86.32**	51.09
$k = 3$ (default)	58.17	83.28	**55.24**
$k = 5$	**67.47**	75.89	54.02

Table 12: Analysis of pre-retrieval, judgment in the fact-guided decomposition module, and final accuracy under different top-k settings.

facts for guidance during the pre-retrieval step. We use the recall of edited facts as the evaluation metric. This metric represents the proportion of all facts required to be edited for a complex question that are successfully selected by LLMs during the pre-retrieval step. Overall, as the base LLMs' reasoning ability improves, recall also increases, which is consistent with expectations.

However, a higher recall does not always translate into better overall performance. The reasons may be complex, but through analysis, we find that one possible reason could be the varying degrees of sensitivity to counterfactuals across different base LLMs. Some models may treat edited facts with excessive skepticism, which can hinder their ability to generate appropriate subquestions. For instance, given the question "What is the capital of the country where American Ninja Warrior originated?", the edited fact "The capital of the United Kingdom is Angri." may trigger unnecessary doubt. A counterfact-sensitive model might respond with a subquestion like "Is Angri the correct capital of the United Kingdom?", disrupting the intended reasoning path.

To explore this, we conduct a simple statistical analysis of some common forms of doubt expressions, such as "(Note: ***)" or "Is *** correct?", and count their occurrences in model outputs (#Doubts). In Table 10, the models that exhibit more such expressions tend to perform worse, suggesting that increased sensitivity to counterfactuals may negatively impact reasoning effectiveness.

J Effectiveness of Fact-Guided Decomposition on Middle-Hop Edits

We evaluate the effectiveness of fact-guided decomposition on middle-hop edits. We first identify the number of cases in which edited facts occur in a middle hop across different datasets. Specifically, there are 411 such 4-hop cases in MQuAKE-2002, and 429 such 4-hop cases in MQuAKE-hard. We then evaluate model performance with and without fact-guided decomposition on these cases, reporting both final accuracy (Acc) and the recall of edited facts (Recall), as defined in Section 5.4. Recall measures the proportion of all required edited facts for a given multi-hop question that are successfully captured during reasoning. All experiments are conducted using LLaMa3-8B-Instruct. The results are shown in Table 11. We observe that applying fact-guided decomposition significantly improves both the recall of edited facts and the final accuracy, demonstrating its utility in handling middle-hop edits.

K Analysis of Fact-Guided Decomposition Module Across Different Numbers of Pre-retrieved Facts

We conduct an analysis to better understand the fact-guided decomposition module by varying the number of pre-retrieved facts. Specifically, we compare the performance of the module under different top-k pre-retrieval settings ($k = 1, 3, 5$) on the MQuAKE-2002 dataset using LLaMa3-8B-Instruct. For each setting, we report three metrics:

- **Recall in pre-retrieval**: the recall of required edited facts retrieved in the pre-retrieval step;
- **Acc in judgment**: whether the model accurately determines the correct guidance behavior. If the required edited fact is not retrieved in the pre-retrieval step, the model should output 0 (no guidance); otherwise, it should correctly select the required edited fact;
- **Acc**: the overall performance of the model.

As shown in Table 12, increasing k improves the recall of edited facts during the pre-retrieval step, but also makes it more challenging for the model to make correct guidance decisions. To achieve better overall performance, the choice of k should be made with a trade-off between recall and judgment accuracy.

Judgment Prompt

```
You are a question decomposition assistant and are given three pieces of knowledge.
There is a complex question that needs to be solved step by step now.
Your task is to assess whether any one of these knowledge items are helpful for guiding the first step of
decomposition to solve the complex question.
To be noted that this help is aimed at the first step of question decomposition, which can be used as a starting
point to further decompose the complex question.
This guidance can either be:
a. Explicit guidance: The knowledge item is directly relevant to the question and can be directly applied to help
decompose the question.
b. Implicit guidance: The knowledge item may not be directly relevant to the question, but its structure, type,
or natural language formulation could aid in decomposing the question.

Below I give 3 examples to further explain:
[3 in-context demonstrations abbreviated]

Now we provide you with a complex question and three pieces of knowledge.
Please determine if the following three pieces of knowledge are helpful for the first step decomposition of the
question.
It should be noted that the assistance is needed for the first step of problem decomposition, which can be used
as a starting point to further decompose the complex question.
If there is any knowledge provided that can help decompose, please return the ID.
If none exist, return 0.
Please return the knowledge id you choose strictly in the following format:
Output:
**<knowledge id>**

Question: <<QUESTION>>
Knowledge:
1. <<KNOWLEDGE1>>
2. <<KNOWLEDGE2>>
3. <<KNOWLEDGE3>>
Output:
```

Figure 7: Prompt for judging helpful knowledge used for question decomposition.

Question Rewriting Prompt

```
Given a complex multi-hop original question, you are provided with additional information consisting of a
subquestion relevant to solving the original complex question, along with its corresponding answer.
Your task is now to rewrite the original question based on the provided information(including the subquestion
and its answer).
Please ensure that your reformulation only takes into account the original question and the given information,
leading to a more informed rephrasing of the original question.

Important:
Do not introduce any external knowledge or assumptions beyond what is provided, and rely solely on the given
information.
Please Note: The answers of the subquestion given may conflict with the facts you already know, you task is to
strictly follow the information provided to finish your task!!!!

To be noted:
Please output your rewritten question in the following format:
**<the rewritten original question>**

For example:
[3 in-context demonstrations abbreviated]

Please complete the following questions:

Complex Multi-hop Original Question: <<QUESTION>>
Sub-Question: <<SUBQUESTION>>
The answer to the sub-question: <<ANSWER TO THE SUBQUESTION>>
Output:
```

Figure 8: Prompt for question rewriting.

Subquestion Generation Prompt

```
Question: What is the capital city of the country of citizenship of Ivanka Trump's spouse?
Subquestion: Who is Ivanka Trump's spouse?
Generated answer: Ivanka Trump's spouse is Jared Kushner.
Subquestion: What is the country of citizenship of Jared Kushner?
Generated answer: Jared Kushner is a citizen of Canada.
Subquestion: What is the capital city of Canada?
Generated answer: The capital city of Canada is Ottawa.
Final answer: Ottawa

Question: Who is the head of state of the country where Rainn Wilson holds a citizenship?
Subquestion: What is the country of citizenship of Rainn Wilson?
Generated answer: Rainn Wilson is a citizen of Croatia.
Subquestion: What is the name of the current head of state in Croatia?
Generated answer: The name of the current head of state in Croatia is Kolinda Grabar-Kitarović.
Final answer: Kolinda Grabar-Kitarović

Question: Who is the spouse of the head of state in United States of America?
Subquestion: Who is the head of state in United States of America?
Generated answer: The head of state in United States of America is Joe Biden.
Subquestion: Who is the spouse of Joe Biden?
Generated answer: The spouse of Joe Biden is Jill Biden.
Final answer: Jill Biden

Question: On which continent is the country of citizenship of the founder of the manufacturer of iPhone 5
situated?
Subquestion: Which company is iPhone 5 produced by?
Generated answer: The company that produced iPhone 5 is Iveco.
Subquestion: Who is the founder of Iveco?
Generated answer: Iveco was founded by Giovanni Agnelli.
Subquestion: What is the country of citizenship of Giovanni Agnelli?
Generated answer: Giovanni Agnelli is a citizen of Niger.
Subquestion: On which continent is Niger situated?
Generated answer: Niger is situated on Africa.
Final answer: Africa
```

Figure 9: Prompt for question decomposition without guidance.

Subquestion Generation Prompt with the Reference Question

```
Question: What is the capital city of the country of citizenship of Ivanka Trump's spouse?
The reference question for the next Subquestion: Who is Ivanka Trump married to?
Subquestion: Who is Ivanka Trump's spouse?
Generated answer: Ivanka Trump's spouse is Jared Kushner.
The reference question for the next Subquestion: What is the country of citizenship of Jared Kushner?
Subquestion: What is the country of citizenship of Jared Kushner?
Generated answer: Jared Kushner is a citizen of Canada.
The reference question for the next Subquestion: What is the capital city of USA?
Subquestion: What is the capital city of Canada?
Generated answer: The capital city of Canada is Ottawa.
Final answer: Ottawa

Question: Who is the head of state of the country where Rainn Wilson holds a citizenship?
The reference question for the next Subquestion: What is the country of citizenship of Johnson?
Subquestion: What is the country of citizenship of Rainn Wilson?
Generated answer: Rainn Wilson is a citizen of Croatia.
The reference question for the next Subquestion: Who is the head of state in Croatia?
Subquestion: What is the name of the current head of state in Croatia?
Generated answer: The name of the current head of state in Croatia is Kolinda Grabar-Kitarović.
Final answer: Kolinda Grabar-Kitarović

Question: Who is the spouse of the head of state in United States of America?
The reference question for the next Subquestion: What is the name of the chief of state of United States of America?
Subquestion: Who is the head of state in United States of America?
Generated answer: The head of state in United States of America is Joe Biden.
The reference question for the next Subquestion: Who is Joe Biden married to?
Subquestion: Who is the spouse of Joe Biden?
Generated answer: The spouse of Joe Biden is Jill Biden.
Final answer: Jill Biden

Question: On which continent is the country of citizenship of the founder of the manufacturer of iPhone 5 situated?
The reference question for the next Subquestion: Which company is Volvo P1800 produced by?
Subquestion: Which company is iPhone 5 produced by?
Generated answer: The company that produced iPhone 5 is Iveco.
The reference question for the next Subquestion: Who founded Iveco?
Subquestion: Who is the founder of Iveco?
Generated answer: Iveco was founded by Giovanni Agnelli.
The reference question for the next Subquestion: What is the country of citizenship of Mike?
Subquestion: What is the country of citizenship of Giovanni Agnelli?
Generated answer: Giovanni Agnelli is a citizen of Niger.
The reference question for the next Subquestion: Which continent is Niger located in?
Subquestion: On which continent is Niger situated?
Generated answer: Niger is situated on Africa.
Final answer: Africa
```

Figure 10: Prompt for question decomposition with guidance from the edited fact level.