StepKE: Stepwise Knowledge Editing for Multi-hop Question Answering

Jaewook Lee* **Dahyun Jung*** **Heuiseok Lim†**
Department of Computer Science and Engineering, Korea University
{jaewook133,dhaabb55,limhseok}@korea.ac.kr

Abstract

Knowledge editing aims to update Large Language Models (LLMs) with new information without costly retraining. However, consistently reflecting these updates in complex multi-hop Question Answering (QA), which demands reasoning over interconnected facts, is challenging. Many existing methods overlook the interplay with pre-existing knowledge, leading to inconsistent edit propagation. To overcome this, we introduce StepKE (Stepwise Knowledge Editing for Multi-hop QA), a novel framework for robustly integrating edited and existing knowledge for coherent multi-hop reasoning. StepKE uniquely decomposes multi-hop questions into sequential single-hop subquestions, retrieving relevant facts (both edited and pre-existing) from an external knowledge graph for each step. It employs context-aware prompting with prior reasoning history and fine-tuning for precise edit propagation. This systematic integration enables effective stepwise reasoning. Experiments show StepKE generates significantly more accurate and consistent responses than baselines, showcasing strong knowledge editing and integration in multi-hop QA.

1 Introduction

Large Language Models (LLMs) often struggle with outdated or new knowledge despite their advanced language capabilities. Knowledge editing, updating LLMs without costly retraining, is thus a critical research area (Yao et al., 2023; Wang et al., 2024a,b; Zhang et al., 2024b). While progress has been made, effectively ensuring modified knowledge is consistently reflected in complex, multi-hop Question Answering (QA) remains challenging. Multi-hop QA requires reasoning over interconnected knowledge, making seamless integration of new edits with pre-existing knowledge essential (Zheng et al., 2023; Meng et al., 2023b; Li et al., 2024a; Fang et al., 2024).

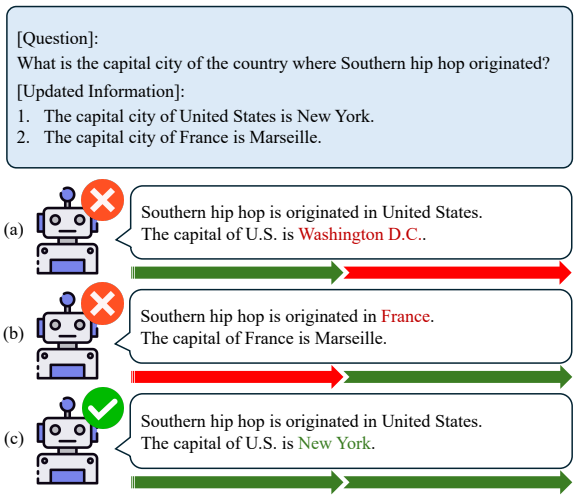


Figure 1: Examples of knowledge editing for multi-hop QA. (a) An example of edit failure. (b) An example of alignment failure caused by an error in answering about unedited knowledge. (c) An example of a successful alignment of edited knowledge with existing knowledge.

Existing research on knowledge editing for multi-hop QA often assumes that only edited knowledge is used in responses. This overlooks interactions with the model's vast inherent background knowledge, potentially leading to inconsistent application of learned modifications during multi-step reasoning and resulting in incomplete or contradictory answers (Chen et al., 2019; Lan et al., 2021; Mavi et al., 2022; Yang et al., 2024). Figure 1 shows errors in multi-hop QA when edited knowledge is not reflected or when necessary existing knowledge is not adequately incorporated. For instance, methods like RAE (Shi et al., 2024), despite considering both knowledge types, can struggle to simultaneously retrieve and integrate multiple items for complex multi-hop questions, limiting accurate and consistent reasoning chains. Others like

*Equal contribution.
†Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 7752–7765
November 4-9, 2025

PokeMQA (Gu et al., 2024) and MeLLo (Zhong et al., 2024) have explored question decomposition or external knowledge graphs, but robustly connecting edited with existing knowledge in the multi-hop reasoning flow remains a challenge, risking information distortion or omission during propagation.

To address these limitations, we propose **StepKE** (**Step**wise **K**nowledge **E**diting for Multi-hop QA), a novel framework enhancing the integration and consistent application of edited knowledge in multi-hop reasoning. StepKE's contribution lies in its unique, systematic approach synergistically combining key components. Our framework meticulously decomposes complex multi-hop questions into simpler, single-hop sub-questions. For each, StepKE retrieves relevant knowledge—both pre-existing and newly edited facts—from an external knowledge graph, ensuring access to current and pertinent information at each step.

A key aspect of StepKE is its context-aware prompting and an LTE-based fine-tuning approach (Zhang et al., 2025). Context-aware prompting ensures coherent reasoning by integrating prior knowledge and reasoning outcomes into subsequent steps. LTE-based fine-tuning then enables accurate reflection of edits while preserving existing knowledge, ensuring precise edit propagation. This synergy of decomposition, knowledge graph augmentation, context-aware prompting, and targeted fine-tuning allows StepKE to sequentially apply linked knowledge, maximizing reasoning consistency and accuracy after edits.

Our experiments show StepKE significantly surpasses existing baselines in accuracy and efficiency. Ablation studies confirm each component's meaningful contribution, underscoring our structured, stepwise approach for reliable multi-hop QA in dynamic knowledge settings. Our main contributions are:

- A novel framework, StepKE, for consistent multi-hop QA that effectively integrates edited and pre-existing knowledge by sequentially applying retrieved, relevant knowledge to decomposed sub-questions.
- A context-aware prompting strategy incorporating reasoning history and retrieved knowledge at each step to enhance multi-hop reasoning coherence and accuracy.
- Demonstration through multi-hop QA benchmarks that StepKE achieves superior editing accuracy by effectively managing knowledge interconnectivity.

2 Related Work

Our work on StepKE builds upon research in knowledge editing for LLMs and multi-hop QA.

2.1 Knowledge Editing in LLMs

Efficiently updating LLM knowledge without full retraining is crucial. Approaches range from direct weight modification (e.g., MEND (Mitchell et al., 2022), ROME (Meng et al., 2023a)) to methods employing compensatory layers or external memory. LTE (Jiang et al., 2024), for instance, fine-tunes a hypernetwork for single-hop factual edits. While promising for atomic edits, ensuring consistent propagation in multi-step reasoning is challenging. StepKE uses an LTE-based fine-tuning for precise edits within a broader multi-hop consistency framework.

2.2 Multi-hop Question Answering

Multi-hop QA requires reasoning over multiple pieces of evidence. Traditional methods involve passage retrieval for a reader model. Recent approaches leverage LLMs' reasoning via techniques like Chain-of-Thought (CoT) (Wei et al., 2022) or question decomposition. KELDar (Li et al., 2024b), for example, decomposes questions and retrieves facts from knowledge graphs. While sharing question decomposition, StepKE distinctively focuses on integrating and propagating edited knowledge cohesively with existing facts at each step.

2.3 Knowledge Editing for Multi-hop QA

This intersection presents challenges in harmonizing updates with existing knowledge throughout long reasoning chains. RAE (Shi et al., 2024) retrieves a relevant subgraph and seeks a fact chain maximizing mutual information. However, its beam search can find local optima, and subgraph extraction may struggle with precise sequential integration for complex queries, potentially including irrelevant or omitting essential facts. StepKE employs a structured, sequential approach, decomposing questions into single-hop queries and retrieving targeted knowledge, mitigating subgraph search complexities. PokeMQA (Gu et al., 2024) and MeLLo (Zhong et al., 2024) (provider of the MQuAKE benchmark) also address this. While decomposing questions or using external knowledge, these methods can be limited in robustly connecting

new edits with existing knowledge during sequential multi-hop reasoning. StepKE retrieves relevant edited and existing facts and explicitly incorporates reasoning history and retrieved knowledge into subsequent prompts via its context-aware mechanism, enabling controlled information propagation. GMeLLo (Chen et al., 2024) relies on SPARQL for knowledge graph access, which can be limiting. StepKE's semantic similarity-based retrieval offers more flexibility.

A key characteristic of StepKE is the systematic integration of knowledge editing with the demands of multi-hop reasoning. Unlike prior works focusing on isolated aspects or struggling with edited information flow, StepKE orchestrates decomposition, retrieval of existing/edited facts, context-aware prompting, and LTE-based editing. This ensures edited knowledge is actively and accurately used throughout the reasoning chain. Ablation studies and baseline comparisons confirm this integrated strategy's crucial role in robust multi-hop knowledge editing.

3 Preliminary

In this section, we formalize knowledge editing for multi-hop QA, emphasizing a realistic setting that accounts for the alignment between existing and edited knowledge using a knowledge graph.

3.1 Knowledge Representation and Editing

We define factual knowledge as a triple (s, r, o), where s is the subject, r is the relation, and o is the object, forming a structured representation of information (Meng et al., 2023a,b). Knowledge editing aims to update a model $\mathcal{M}$'s knowledge. Specifically, it often involves modifying an existing knowledge triple $k = (s, r, o)$ into an edited triple $k^* = (s, r, o^*)$, where the object o is updated to o^*, while s and r remain unchanged. Let $\mathcal{K}_{orig}$ denote the original knowledge graph representing pre-existing facts, and $K_{edit} = \{k_1^*, k_2^*, \ldots, k_M^*\}$ be a set of M new factual edits. We define an external knowledge graph $\mathcal{K}^+ = \mathcal{K}_{orig} \oplus K_{edit}$, where $\oplus$ indicates that K_{edit} is integrated into $\mathcal{K}_{orig}$, potentially overriding conflicting facts. Our framework aims to retrieve necessary knowledge for answering questions from this $\mathcal{K}^+$.

3.2 Multi-hop Question Answering

Multi-hop QA is the task of answering a question Q that requires reasoning over a chain of multiple factual triples $C = \{(s_i, r_i, o_i)\}_{i=1}^n$ (Zhang et al., 2024a; Wu et al., 2024; Gu et al., 2024). Given Q, the model must sequentially reason over these interconnected facts to generate a final answer A. In the context of KE for multi-hop QA, if a fact (s_i, r_i, o_i) within the required reasoning chain is edited to (s_i, r_i, o_i^*), the subsequent reasoning steps must consistently reflect this change to derive the correct answer. For instance, the next fact in the chain might need to be identified as $(o_i^*, r_{i+1}, o_{i+1})$ or similar, ensuring the edited knowledge propagates correctly.

3.3 Problem Statement

The problem we address is knowledge editing for multi-hop QA within an integrated knowledge environment. Given an LLM, a multi-hop question Q, an original knowledge graph $\mathcal{K}_{orig}$, and a set of edits K_{edit} (collectively forming an external knowledge base $\mathcal{K}^+$), the primary goal is to generate an answer A that accurately and consistently reflects the updated knowledge throughout the entire multi-hop reasoning process. This formulation extends beyond constrained settings that typically consider only isolated edited facts, moving towards a more realistic scenario where edited and pre-existing knowledge are interconnected and must be navigated coherently.

Achieving effective knowledge editing in this complex multi-hop setting necessitates satisfying two key requirements. First, Local Consistency ensures that the edited model accurately incorporates any modified knowledge $k^* \in K_{edit}$ when directly queried. Second, Global Consistency demands that the edited knowledge K_{edit} is seamlessly integrated with the original knowledge $\mathcal{K}_{orig}$ without introducing logical contradictions. This preserves coherence throughout complex multi-hop reasoning that leverages facts from both K_{edit} and $\mathcal{K}_{orig}$.

4 Stepwise Knowledge Editing

As illustrated in Figure 2, we propose StepKE, a framework designed to effectively incorporate edited knowledge into multi-hop QA while maintaining alignment with existing knowledge. StepKE is not merely a linear sequence of steps, but rather a synergistic integration of four key components: 1) Question Decomposition for structuring the reasoning process, 2) Knowledge Augmented Generation for grounding each reasoning step with rel-

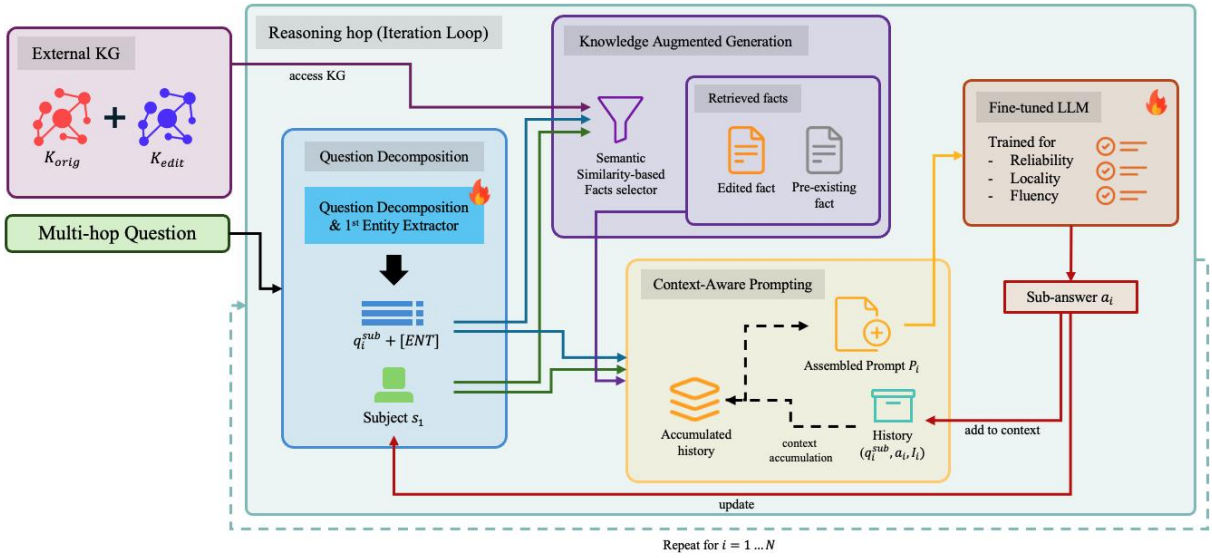


Figure 2: Overview of the StepKE framework. StepKE processes a multi-hop question with an external knowledge graph (containing original K_{orig} and edited K_{edit} knowledge) by first decomposing it into sub-questions and extracting the initial subject s_i. The core Reasoning hop (Iteration Loop) then handles each sub-question q_i^{sub} through: (1) Knowledge Augmented Generation, retrieving relevant edited/pre-existing facts I_i from the knowledge graph. (2) Context-aware Prompting, constructing prompt P_i using q_i^{sub}, s_i, I_i, and crucially, accumulated history from prior hops. (3) A Fine-tuned LLM generating sub-answer a_i from P_i. This iterative process, where a_i informs s_{i+1} and enriches subsequent context, repeats to derive the final answer, ensuring consistent and accurate edit propagation via synergistic component interaction.

evant facts, 3) Fine-tuning for Knowledge Editing to ensure accurate application of edits, and 4) Context-aware Prompting to maintain coherence across multiple reasoning hops. The first two components define the primary inference pipeline for each hop, while the latter two are crucial strategies for enabling reliable knowledge editing within this pipeline.

4.1 Question Decomposition

Given a complex multi-hop question Q, StepKE first decomposes it into an ordered sequence of simpler, single-hop sub-questions $(q_1^{sub}, q_2^{sub}, \ldots, q_N^{sub})$. Simultaneously, the subject s_1 of the first sub-question q_1^{sub} is extracted. This decomposition refines the reasoning chain step-by-step, providing a well-defined structure for subsequent processes and improving overall reasoning accuracy by minimizing the propagation of entity identification errors (Gu et al., 2024). StepKE is trained to jointly generate these multiple sub-questions and s_1 using a prompt template $\mathcal{P}$ based on Q. The objective function is the language modeling loss:

$$\mathcal{L}_{dec}(\theta) = -\sum_{t=1}^{T} \log p_\theta(q_1^{sub}, \ldots, q_N^{sub}, s_1 | \mathcal{P}, Q_t), \quad (1)$$

where θ represents the model parameters, and T is the number of training instances. In the generated sub-questions q_i^{sub} for $i > 1$, the subject entity (which is the answer to q_{i-1}^{sub}) is replaced with a special placeholder token '[ENT]'. This guides the QA model to focus on simplified single-hop reasoning for each sub-question.

4.2 Knowledge Augmented Generation

For each decomposed sub-question q_i^{sub} (where the '[ENT]' token is resolved to the answer a_{i-1} from the previous step, and $s_i = a_{i-1}$ for $i > 1$, while s_1 is directly extracted), StepKE retrieves relevant factual knowledge from the external knowledge graph $\mathcal{K}^+$ (which integrates both $\mathcal{K}_{orig}$ and K_{edit}). Specifically, for q_i^{sub} with its identified subject s_i, we extract candidate triples from $\mathcal{K}^+$ where s_i is the head entity. These triples may include both facts directly relevant to answering q_i^{sub} and irrelevant ones, encompassing both pre-existing and potentially edited knowledge. To select the most pertinent facts I_i for q_i^{sub}, we employ a sentence-

transformer model (Reimers, 2019) to compute the cosine similarity between the sentence embeddings of each candidate fact and q_i^{sub}, selecting the top-k most similar facts.

The iterative answering process is as follows: For the first sub-question q_1^{sub} with subject s_1 and its retrieved relevant facts I_1, an answer a_1 is generated using the fine-tuned LLM described in Section 4.3 via the Context-aware Prompt in Section 4.4. This answer a_1 then becomes the subject s_2 by replacing '[ENT]' in the next sub-question q_2^{sub}. The process of retrieving facts I_i for q_i^{sub} with subject $s_i = a_{i-1}$ and generating answer a_i is repeated until the final sub-question q_N^{sub} is answered, yielding the final answer $A = a_N$ to the original multi-hop question Q.

4.3 Fine-tuning for Knowledge Editing

To ensure that the LLM accurately reflects the edited knowledge from K_{edit} (retrieved as part of I_i) when answering each sub-question q_i^{sub}, we fine-tune the base LLM using an approach based by LTE (Jiang et al., 2024). The primary objective is to enable the model to seamlessly integrate edited information with pre-existing knowledge from $\mathcal{K}_{orig}$ (also potentially in I_i), ensuring that edited content is appropriately reflected in the response to q_i^{sub} without negatively impacting unrelated knowledge or reasoning abilities. The fine-tuning process focuses on achieving:

- **Reliability**: The model consistently generates responses reflecting the updated information when queried about edited facts.
- **Locality**: Changes are localized to relevant queries, leaving responses to unedited or unrelated queries intact.
- **Fluency**: Responses maintain natural language structure and coherence while integrating both edited and existing knowledge.

This targeted fine-tuning allows the model to effectively utilize the potentially edited facts I_i provided through the Context-aware Prompt for each step of the multi-hop reasoning.

4.4 Context-aware Prompting

To further enhance the model's reasoning capability across the sequence of interconnected single-hop sub-questions, particularly in the context of knowledge editing, we employ a Context-aware Prompting strategy. As depicted in Figure 3, this strategy

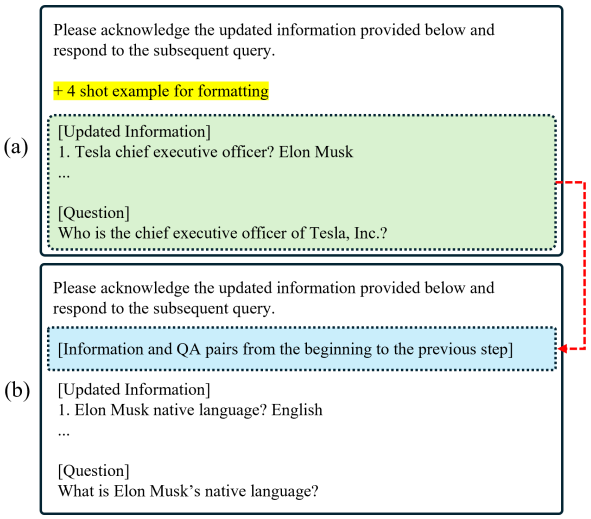


Figure 3: An example of Context-aware Prompting. (a) An example prompt for the first sub-question, including a 4-shot example. (b) An example of the prompt for the second sub-question, where the information and QA pairs from the previous step are accumulated and utilized.

accumulates the generated question-answer pairs (q_j^{sub}, a_j) and the retrieved supporting facts I_j from all previous hops $j < i$ when constructing the prompt for the current sub-question q_i^{sub}. For the first sub-question q_1^{sub}, the prompt includes q_1^{sub}, its subject s_1, the retrieved facts I_1, and a few-shot example for formatting (Figure 3a). From the second sub-question q_2^{sub} onward (Figure 3b), the prompt explicitly includes this accumulated history of previous reasoning steps (e.g., "Given that q_1^{sub} was answered by a_1 using facts I_1, and q_2^{sub} was answered by a_2 using facts I_2, now answer q_3^{sub} based on facts I_3..") before presenting q_i^{sub} and I_i. This explicit contextualization allows the model to:

1. Understand the current sub-question q_i^{sub} within the broader reasoning chain.
2. Leverage previously inferred information and the specific (edited or pre-existing) facts that supported those inferences.
3. Maintain consistency, ensuring that the application of an edit in an earlier step is correctly propagated to subsequent steps.

This structured prompting guides the model to perform consistent and context-aware multi-hop reasoning, akin to fostering a more grounded CoT process (Wei et al., 2022) that is explicitly aware of the knowledge editing context.

Method	LLaMA-3.1-8B			Qwen-2.5-7B		
	MQuAKE-3k	MQuAKE-2002	MQuAKE-hard	MQuAKE-3k	MQuAKE-2002	MQuAKE-hard
Single-hop Editing Methods						
FT	0.0150	0.0180	0.0047	0.0240	0.0150	0.0023
MEMIT	0.0003	0.0015	0.0047	0.0000	0.0005	0.0000
LTE	0.0527	0.0569	0.0186	0.1260	0.1314	0.0816
Multi-hop Editing Methods						
PokeMQA	0.0443	0.1049	0.0443	0.1220	0.1444	0.1492
DeepEdit	0.1167	0.1548	0.0117	0.0660	0.0779	0.0210
RAE	0.6880	0.7707	0.6713	0.6240	0.7133	0.6107
StepKE	**0.8200**	**0.8287**	**0.7389**	**0.8493**	**0.8676**	**0.7365**

Table 1: Experimental results on MQuAKE-3k, MQuAKE-2002, and MQuAKE-hard. We compare StepKE against single-hop and multi-hop baseline methods on the LLaMA-3.1 and Qwen-2.5 models.

5 Experiments

In this section, we present comprehensive experiments to evaluate our proposed methodology, StepKE. We aim to demonstrate its effectiveness in answering multi-hop QA in scenarios where both existing and edited knowledge are provided, a setting that reflects more realistic knowledge editing challenges.

5.1 Experimental Setup

Datasets and Evaluation Metrics. We evaluate our proposed method on three benchmark datasets: MQuAKE-3k (Zhong et al., 2024), MQuAKE-2002 (Wang et al., 2024c), and MQuAKE-hard (Wang et al., 2024c). MQuAKE-3k is designed to assess whether a model with edited knowledge can generate correct answers to multi-hop questions involving both edited and unedited facts. MQuAKE-2002 is a refined version of MQuAKE-3k, where conflicts between new and pre-existing knowledge have been resolved, focusing on the model's ability to use edited facts in new contexts. MQuAKE-hard is a more challenging dataset requiring reasoning over non-overlapping 4-hop facts, testing deeper reasoning capabilities post-editing. Statistical details of each dataset are provided in Table 4. To evaluate whether the model successfully integrates edited facts into its responses and answers the multi-hop questions correctly, we use accuracy as the primary evaluation metric.

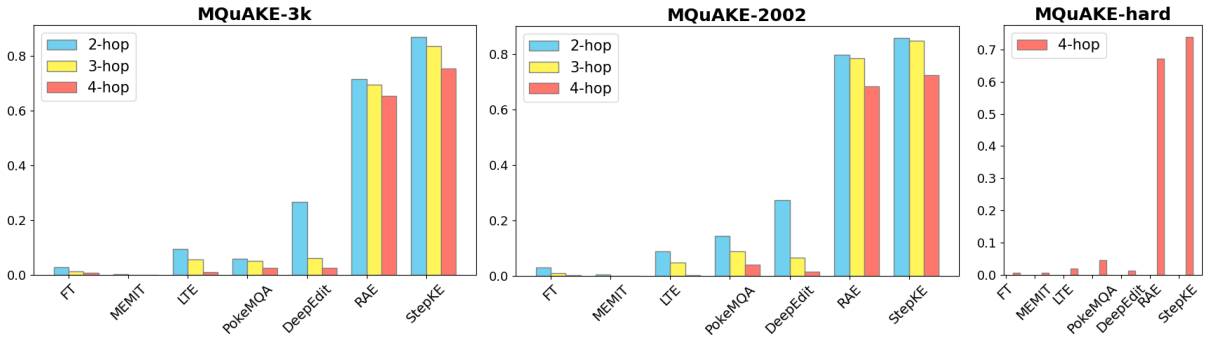
Baselines. We compare StepKE against representative single-hop-based knowledge editing methods (FT (Zhu et al., 2020), MEMIT (Meng et al., 2023b), LTE (Jiang et al., 2024)) and state-of-the-art multi-hop-based methods (PokeMQA (Gu et al., 2024), DeepEdit (Wang et al., 2024c), RAE (Shi et al., 2024)). For detailed descriptions of baselines, please refer to their respective papers.

Implementation Details. We conduct experiments using LLaMA-3.1 8B (Llama Team, 2024) and Qwen-2.5 7B (Qwen et al., 2025). Model checkpoints are 'meta-llama/Llama-3.1-8B-Instruct' and 'Qwen/Qwen2.5-7B-Instruct' from HuggingFace. All experiments are performed using two NVIDIA A100-80GB GPUs. For FT and our LTE-based fine-tuning in StepKE, following Jiang et al. (2024), we set epochs to 3, learning rate to 2×10^{-5}, and use AdamW. For MEMIT, following Wang et al. (2024a), we update layers $\{4, 5, 6, 7, 8\}$. We sample 10K times from WikiText for covariance matrix C estimation and set $\lambda = 15000$. We use official codebases for PokeMQA, DeepEdit, and RAE where available. Further details are in Appendix B.

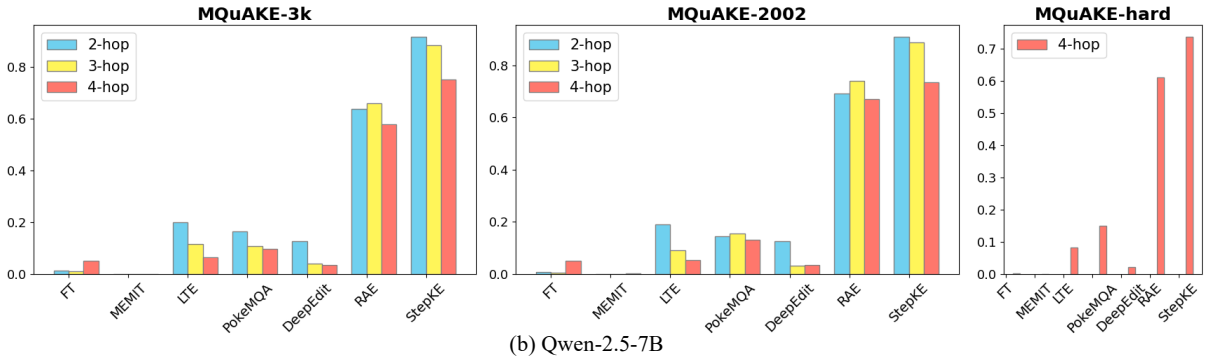
5.2 Main Results

Table 1 presents the primary experimental results on the MQuAKE-3k, MQuAKE-2002, and MQuAKE-hard benchmarks. StepKE consistently outperforms other knowledge editing approaches across all datasets and base models, particularly in effectively handling scenarios that require integration of both edited and pre-existing knowledge. Compared to RAE, a strong baseline that incorporates retrieval-based pruning, StepKE demonstrates superior editing capabilities. For instance, on MQuAKE-3k with LLaMA-3.1, RAE achieves 0.688, while StepKE reaches 0.82. Similarly, with Qwen-2.5, RAE scores 0.624, whereas StepKE achieves 0.8493. These results underscore StepKE's effectiveness in integrating new knowledge for accurate multi-hop responses by effectively capturing knowledge interconnectivity.

Furthermore, when compared to single-hop editing methods like FT, MEMIT, and LTE, StepKE



(a) LLaMA-3.1-8B



(b) Qwen-2.5-7B

Figure 4: Accuracy on 2, 3, and 4-hop questions in MQuAKE-3k, MQuAKE-2002 and MQuAKE-hard.

achieves significantly higher performance. Notably, while LTE is a strong single-hop editing approach, its direct application to multi-hop scenarios is limited (e.g., 0.0527 on MQuAKE-3k with LLaMA-3.1). StepKE, which incorporates an LTE-based fine-tuning strategy within its structured multi-hop framework, achieves 0.82 on the same setup. This substantial improvement highlights that StepKE's comprehensive approach—combining question decomposition, knowledge augmentation, and context-aware prompting with targeted fine-tuning—is far more effective than applying single-hop editing techniques in isolation to complex multi-hop tasks. This demonstrates that StepKE's design successfully addresses the unique challenges of propagating edits through multi-step reasoning chains. Multi-hop specific methods like PokeMQA and DeepEdit also show limitations in these complex, large-scale knowledge environments compared to StepKE.

5.3 Analysis by Number of Hops

Figure 4 illustrates the performance breakdown for 2-hop, 3-hop, and 4-hop questions on the MQuAKE datasets. StepKE consistently achieves the highest performance across all hop counts and datasets. Even when compared to RAE, which is relatively robust for questions with 3 or more hops, StepKE maintains superior editing performance, particularly excelling on the more complex MQuAKE-hard dataset, which consists exclusively of 4-hop queries, when compared to other methodologies. In contrast, DeepEdit exhibits a sharp decline in performance as the reasoning complexity increases with the number of hops, and RAE struggles even with 2-hop questions. StepKE, however, maintains consistently high and stable performance across varying hop counts. This stability suggests that StepKE's methodology of decomposing questions and sequentially retrieving and integrating both edited and existing knowledge is effective even as the reasoning chain lengthens and editing propagation becomes more critical.

Method	MQuAKE-3k	MQuAKE-2002	MQuAKE-hard
StepKE	**0.8200**	**0.8287**	0.7389
w/o QD	0.0527	0.0569	0.0186
w/o KG	0.4693	0.6433	**0.9347**
w/o CP	0.7313	0.7253	0.6084
w/o LTE	0.7340	0.8052	0.7203

Table 2: Ablation study on StepKE. QD is Question Decomposition, KG is Knowledge Graph, CP is Context-aware Prompting, and LTE is Fine-tuning for Knowledge Editing.

Retriever	top-k	MQuAKE-3k	MQuAKE-2002	MQuAKE-hard
StepKE	1	**0.8200**	**0.8287**	**0.7389**
	3	0.7903	0.8062	0.4312
	5	0.7173	0.8082	0.4359
StepKE *w/* BM25	1	0.5610	0.5614	0.3800
	3	0.6813	0.6778	0.3473
	5	0.7290	0.7343	0.4592

Table 3: Performance comparison of different retrievers on MQuAKE datasets.

5.4 Ablation Study

To evaluate the contribution of each key component in StepKE, we conducted an ablation study, presented in Table 2, using LLaMA-3.1 on the MQuAKE datasets.

- **w/o Question Decomposition (QD)**: Removing the QD module, which structures the multi-hop task into manageable single-hop questions, causes a substantial performance drop. This underscores that QD is critical for StepKE to effectively stage the reasoning process and integrate edited knowledge in a controlled, stepwise manner. The generalizability of our QD module has also been explored with few-shot learning and on other datasets like 2WikiMultihopQA (Ho et al., 2020) (see Appendix D for details). Furthermore, an analysis aimed at mitigating errors that can arise during the QD process is presented in Appendix E.

- **w/o Knowledge Graph (KG)**: Omitting the KG retrieval step, which provides relevant edited and pre-existing facts, significantly degrades performance. This highlights the importance of grounding each reasoning step with explicit knowledge from $\mathcal{K}^+$. The impact on MQuAKE-hard is less pronounced as all necessary facts are often directly provided as edits, as shown in Table 4.

- **w/o Context-aware Prompting (CP)**: Removing CP, which provides historical reasoning context to subsequent steps, also leads to a noticeable performance decline, especially for more complex multi-hop questions. This confirms CP's role in maintaining reasoning consistency and facilitating the propagation of edited information.

- **w/o LTE-based Fine-tuning (LTE)**: Excluding the LTE-based fine-tuning phase, designed to make the model adept at reflecting provided facts, results in a performance drop. This indicates that this targeted training is crucial for ensuring facts from I_i are accurately incorporated into responses at each hop.

These results collectively demonstrate that each component of StepKE plays a vital role, and their synergistic integration is key to the framework's overall effectiveness in multi-hop knowledge editing. The careful orchestration of these components addresses the core challenge of reliably using edited knowledge in extended reasoning.

5.5 Performance by Retrieval Methods

Table 3 compares the impact of different retrieval strategies for selecting facts I_i for each sub-question q_i^{sub}. We compare our semantic similarity-based approach with BM25, varying the number of retrieved facts (top-k). Our dense embedding-based semantic similarity approach generally performs better, especially with smaller k, indicating that high semantic relevance between the sub-question and the retrieved facts is crucial. BM25, relying on lexical matching, shows improved performance with larger k but can be limited by synonyms or paraphrasing. These findings suggest that effective semantic retrieval is beneficial, and further improvements might be gained by incorporating re-ranking mechanisms, such as cross-encoders, which showed promise in preliminary experiments for mitigating error propagation (see Appendix C).

6 Conclusion

In this paper, we introduce StepKE, a framework that enhances multi-hop knowledge editing by integrating both existing and updated facts. StepKE ensures stepwise reasoning consistency through question decomposition and context-aware prompting, leading to more accurate updates. On MQuAKE benchmarks, StepKE outperforms prior methods, effectively addressing knowledge misalignment and reasoning inconsistencies. Moreover, StepKE achieves higher accuracy with a lower average execution time, making it both efficient and scalable. Overall, StepKE provides a robust foundation for multi-hop knowledge editing, offering a more structured and computationally efficient way to integrate new knowledge into large language models.

Limitations

While StepKE has been extensively evaluated across multiple benchmark datasets and knowledge editing scenarios, several limitations remain. First, StepKE primarily targets multi-hop reasoning where single-hop inferences are sequentially connected. However, real-world multi-hop questions often involve parallel reasoning or conditional dependencies, which require further generalization of the Question Decomposition and Alignment techniques. Second, StepKE's stepwise inference method introduces efficiency trade-offs. While it enhances reasoning consistency, it may also increase inference latency compared to end-to-end methods. Optimizing the decoding strategy or incorporating beam search techniques may improve efficiency. Finally, StepKE's adaptability to domain-specific knowledge remains limited. Domains like law or medicine require precise fact verification and terminology handling, which StepKE does not explicitly incorporate. Domain-specific extensions and fact-checking mechanisms could further enhance its applicability. Despite these limitations, StepKE presents a strong foundation for multi-hop knowledge editing and can be further extended to handle more complex reasoning scenarios.

Ethical Statement

The ability to selectively modify specific knowledge within a model carries the risk of malicious use, such as inserting incorrect information or overemphasizing certain perspectives. In this research, we employ datasets (including the MQuAKE series) constructed from a publicly accessible knowledge graph (Wikidata) and question–answer pairs from relevant benchmarks. Our proposed StepKE framework decomposes the reasoning process step by step, incorporating edited knowledge directly into the generated answers. In pursuit of reproducibility, we plan to release detailed implementation components—such as prompt templates, hyperparameters, and algorithmic structures—and to make our code openly available in the future.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (RS-2024-00398115, Research on the reliability and coherence of outcomes produced by Generative AI). This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2021R1A6A1A03045425). This work was supported by the Commercialization Promotion Agency for R&D Outcomes(COMPA) grant funded by the Korea government(Ministry of Science and ICT)(2710086166).

References

Jifan Chen, Shih-ting Lin, and Greg Durrett. 2019. Multi-hop question answering via reasoning chains. *arXiv preprint arXiv:1910.02610*.

Ruirui Chen, Weifeng Jiang, Chengwei Qin, Ishaan Rawal, Cheston Tan, Dongkyu Choi, Bo Xiong, and Bo Ai. 2024. Llm-based multi-hop question answering with knowledge graph integration in evolving environments. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14438–14451.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Xiang Wang, Xiangnan He, and Tat seng Chua. 2024. Alphaedit: Null-space constrained knowledge editing for language models. *Preprint*, arXiv:2410.02355.

Hengrui Gu, Kaixiong Zhou, Xiaotian Han, Ninghao Liu, Ruobing Wang, and Xin Wang. 2024. Pokemqa: Programmable knowledge editing for multi-hop question answering. *Preprint*, arXiv:2312.15194.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Yuxin Jiang, Yufei Wang, Chuhan Wu, Wanjun Zhong, Xingshan Zeng, Jiahui Gao, Liangyou Li, Xin Jiang, Lifeng Shang, Ruiming Tang, Qun Liu, and Wei Wang. 2024. Learning to edit: Aligning llms with knowledge editing. *Preprint*, arXiv:2402.11905.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. A survey on complex knowledge base question answering: Methods, challenges and solutions. *arXiv preprint arXiv:2105.11644*.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024a. Pmet: Precise model editing in a transformer. *Preprint*, arXiv:2308.08742.

Yading Li, Dandan Song, Changzhi Zhou, Yuhang Tian, Hao Wang, Ziyi Yang, and Shuhao Zhang. 2024b. A framework of knowledge graph-enhanced large language model based on question decomposition and atomic retrieval. In *Findings of the Association*

for Computational Linguistics: EMNLP 2024, pages 11472–11485.

Llama Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2022. A survey on multi-hop question answering and generation. *arXiv preprint arXiv:2204.09140*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023a. Locating and editing factual associations in gpt. *Preprint*, arXiv:2202.05262.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023b. Mass-editing memory in a transformer. *Preprint*, arXiv:2210.07229.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. Fast model editing at scale. In *International Conference on Learning Representations*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

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. *Preprint*, arXiv:2403.19631.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. 2024a. Easyedit: An easy-to-use knowledge editing framework for large language models. *Preprint*, arXiv:2308.07269.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024b. Knowledge editing for large language models: A survey. *Preprint*, arXiv:2310.16218.

Yiwei Wang, Muhao Chen, Nanyun Peng, and Kai-Wei Chang. 2024c. Deepedit: Knowledge editing as decoding with constraints. *Preprint*, arXiv:2401.10471.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jian Wu, Linyi Yang, Yuliang Ji, Wenhao Huang, Börje F. Karlsson, and Manabu Okumura. 2024. Gendec: A robust generative question-decomposition method for multi-hop reasoning. *Preprint*, arXiv:2402.11166.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024. Do large language models latently perform multi-hop reasoning? *Preprint*, arXiv:2402.16837.

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. *Preprint*, arXiv:2305.13172.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Liu Yong, and Shen Huang. 2024a. End-to-end beam retrieval for multi-hop question answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1718–1731, Mexico City, Mexico. Association for Computational Linguistics.

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. 2024b. A comprehensive study of knowledge editing for large language models. *Preprint*, arXiv:2401.01286.

Zhuoran Zhang, Yongxiang Li, Zijian Kan, Keyuan Cheng, Lijie Hu, and Di Wang. 2025. Locate-then-edit for multi-hop factual recall under knowledge editing. *Preprint*, arXiv:2410.06331.

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? *Preprint*, arXiv:2305.12740.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. 2024. Mquake: Assessing knowledge editing in language models via multi-hop questions. *Preprint*, arXiv:2305.14795.

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

A Multi-hop QA Dataset Statistics

Table 4 contains the statistics for the MQuAKE benchmark datasets used in our experiments.

B Training Details

Question Decomposition and Subject Extraction Model Figure 5 illustrates the prompt used for training the question decomposition and subject extraction model. This model is designed to decompose a given multi-hop question into sequential sub-questions while extracting the subject from the first sub-question. We employ 'meta-llama/Llama-3.1-8B-Instruct' as the backbone model and train it on the MQuAKE-CF. The training is conducted with the following hyperparameters: the number of epochs is set to 1, the batch size is 32, and the learning rate is 5×10^{-6}. During training, the model receives a structured prompt where the input consists of a multi-hop question. The expected output includes the first sub-question derived from the original query, followed by subsequent sub-questions where the main subject is replaced with `[ENT]` to maintain coherence. Additionally, the model extracts the subject from the first sub-question. This decomposition approach enables the model to perform stepwise reasoning, improving knowledge retrieval and facilitating accurate multi-hop QA.

Fine-tuning for Knowledge Editing We fine-tune the base model following LTE (Jiang et al., 2024). Specifically, we use the AdamW optimizer with a learning rate of 2×10^{-5} and train for 3 epochs. During inference, we set the temperature to 1.0, the number of beams to 5, and the number of facts retrieved based on semantic similarity (top-k) to 1.

C Error Analysis

Figure 6 shows the analysis of error types that occurred when StepKE generated incorrect answers on the MQuAKE dataset. The analysis reveals that errors can be classified into four types (E1, E2, E3, and E4), with similar trends observed across the three datasets (MQuAKE-2002, MQuAKE-3k, and MQuAKE-hard). E1 refers to cases where the subject of the first sub-question was incorrectly extracted. This type of error occurred relatively infrequently. E2 occurs when a fact related to a previous answer could not be found in the external knowledge graph, which is one of the primary reasons for the model's failure to retrieve the correct answer due to a lack of information. E3 represents cases where the target entity's Wikidata ID is missing from the knowledge graph, resulting from database incompleteness. E4 is the most frequent error type, where mistakes in intermediate steps propagate to later sub-questions, ultimately leading to an incorrect final answer.

The dominance of E4 errors suggests that error propagation in multi-hop reasoning significantly affects performance. This indicates that mistakes in earlier reasoning stages can accumulate, severely impacting the final answer. Based on these findings, introducing strategies to minimize errors in question decomposition and intermediate reasoning steps is crucial for enhancing model performance.

You need to divide the given multi-hop question into several sub-questions and extract the subject of the first sub-question. Present the divided sub-questions on every line, and starting with the second sub-question, write the main subject as '[ENT]'.

Question: What is the capital city of the country where Southern hip hop originated?

Subquestion:
Where did Southern hip-hop originate?
What is the capital city of [ENT]?
Subject of 1st Subquestion:
Southern hip hop

Figure 5: An example prompt used for training the question decomposition and subject extraction model. The white background represents the input provided to the model, while the green background indicates the expected model output. The model is trained to decompose a multi-hop question into sequential sub-questions and extract the subject from the first sub-question.

D Evaluation on General Multi-hop QA

To assess StepKE's generalizability beyond MQuAKE and its effectiveness in general multi-hop QA, we experimented on the 2WikiMultihopQA (Ho et al., 2020) dataset, which requires reasoning over multiple documents. We evaluated our LLaMA-3.1-8B StepKE model, focusing on its multi-hop reasoning capabilities. For comparison, we used: (1) the base LLaMA-3.1-8B model without specific retrieval for 2WikiMultihopQA, and (2) the same LLM with context retrieved by a pre-trained cross-encoder based on the multi-hop question. StepKE utilized its MQuAKE-CF-trained Question Decomposition module, semantic retrieval for sub-questions from provided evidence, and Context-aware Prompting. Accuracy (exact match) was the evaluation metric. The results are presented in Table 5. StepKE, with an accuracy of

Benchmark	# Instances	# Edited Facts per Instance	# Hops per Instance	# 2-hop	# 3-hop	# 4-hop
MQUAKE-3k	3,000	2.0	3.0	1,000	1,000	1,000
MQUAKE-2002	2,002	2.2	2.7	966	625	411
MQUAKE-hard	429	4.0	4.0	0	0	429

Table 4: Statistics of MQuAKE datasets.

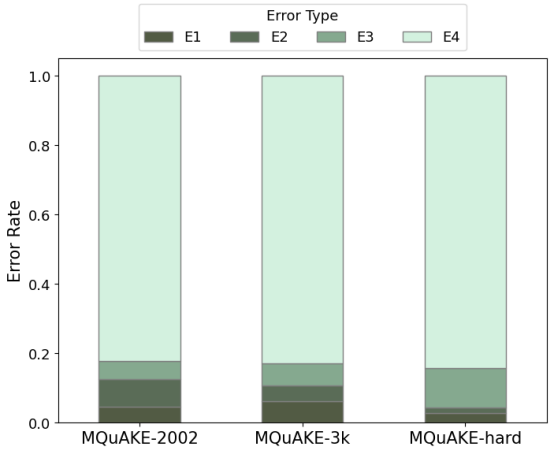


Figure 6: Error analysis on MQuAKE datasets. E1 means that the subject of the first sub-question was incorrectly extracted. E2 means that no facts subject to the previous answer were found in the external graph. E3 means that the target entity's wikidata ID does not exist in the graph. E4 means that the error propagated due to an incorrect answer on the way to the last sub-question (simple incorrect answer).

Method	Accuracy
Base Model (LLaMA-3.1-8B)	0.136
Cross-Encoder Retrieval	0.456
StepKE	**0.598**

Table 5: Performance comparison on the general multi-hop QA dataset 2WikiMultihopQA using LLaMA-3.1-8B.

0.598, significantly outperformed the base model (0.136) and the strong Cross-Encoder retrieval baseline (0.456). This suggests that StepKE's approach of decomposing questions and retrieving targeted evidence for each sub-question is also effective for general multi-hop QA tasks, even with a QD module trained on a different dataset structure. The stepwise reasoning facilitated by StepKE proved more beneficial than providing a larger, less structured context based on the entire multi-hop query. These findings help alleviate concerns about overfitting StepKE's mechanisms to the MQuAKE format and demonstrate its broader applicability and a good level of generalization for multi-hop reasoning challenges.

E Analysis of Error Propagation in Sequential Reasoning

In multi-hop reasoning, a common strategy is to decompose a complex question into a sequence of simpler sub-questions. A key vulnerability of this approach is error propagation, where an error generated in an early reasoning step can compromise the entire process, leading to an incorrect final answer. To investigate and mitigate this issue, we conducted an experiment where we applied a cross-encoder for re-ranking in the facts retrieval stage. In this process, we extract the top-1 fact from the top-5 retrieved facts using a simple error-correction mechanism, which is then used for answering each sub-question. The results of this experiment are detailed in Table 6. Based on the Llama-3.1 8B model, the accuracy on the MQuAKE-CF-3k dataset improved by 5.1 percentage points, on the MQuAKE-2002 dataset by 5.84 percentage points, and on the MQuAKE-hard dataset by 1.86 percentage points. This indicates that StepKE can further reduce errors in each reasoning step by employing a more rigorous retrieval or re-ranking policy.

F Case Study

Table 7 presents a case study comparing StepKE and RAE in answering a complex question requiring multi-hop reasoning. The question asks for the birthplace of the founder of the religion associated with `'Saint Engelbert'`. Since answering this question involves multiple inference steps, the effectiveness of knowledge retrieval and reasoning is critical.

RAE directly retrieves a set of knowledge triples and attempts to infer the answer without explicit intermediate reasoning. As shown in the retrieved knowledge, RAE fails to directly retrieve the correct knowledge required to answer the question and instead constructs an erroneous reasoning chain. This leads to the incorrect fact that Christianity was founded by Adolfo Suárez, ultimately resulting in

the wrong birthplace, Cebreros.

In contrast, StepKE decomposes the reasoning process into step-by-step sub-questions. Each sub-question is answered using retrieved knowledge, ensuring accuracy before proceeding to the next step. This stepwise approach not only directly incorporates the edited knowledge into the answer but also provides a reasoning process based on the connections between knowledge facts, ensuring robustness against incorrect or misleading information. As a result, StepKE successfully derives the correct answer, whereas RAE fails due to incorrect knowledge.

Method	MQUAKE-CF-3k	MQUAKE-2002	MQUAKE-hard
StepKE	0.8200	0.8287	0.7389
StepKE+CE	0.8710 (+0.0510)	0.8871 (+0.0584)	0.7575 (+0.0186)

Table 6: Performance comparison with a cross-encoder re-ranking policy applied for fact retrieval.

QUESTION	What is the birthplace of the founder of the religion associated with Saint Engelbert?
RAE	
Retrieved knowledge	Saint Engelbert position held archbishop. archbishop subclass of bishop. bishop has the religion of Christianity. Christianity was founded by Adolfo Suárez. Adolfo Suárez was born in the city of Cebreros.
Final answer	Cebreros ✗
StepKE	
iteration 1	
Sub-question	Which religion is Saint Engelbert affiliated with?
Retrieved knowledge	Saint Engelbert has the religion of? Methodism (edited knowledge)
Sub-answer	Methodism ✓
iteration 2	
Sub-question	Who founded Methodism?
Retrieved knowledge	Methodism was founded by? John Wesley
Sub-answer	John Wesley ✓
iteration 3	
Sub-question	Which city was John Wesley born in?
Retrieved knowledge	John Wesley was born in the city of? Epworth
Sub-answer	Epworth ✓
Final answer	Epworth ✓

Table 7: Case study comparing StepKE with RAE.