Robust Knowledge Editing via Explicit Reasoning Chains for Distractor-Resilient Multi-Hop QA

Anonymous ACL submission

Abstract

Large language models (LLMs) encode vast amounts of world knowledge but remain static once trained, making timely integration of emerging facts prohibitively expensive via full retraining. Knowledge-editing techniques have thus emerged to inject or overwrite specific facts into LLMs, yet they either over-rely on superficial cues or incur complex, iterative pipelines that collapse under noisy, multi-hop conditions. We introduce **Reason-KE**, an end-to-end reasoning-chain-based editing framework that steers a pretrained LLM through four structured stages—fact acknowledgment, relevance determination, selective application, and final reasoning—to filter distractors in a single pass. Trained on MQuAKE-CF with up to four irrelevant facts, Reason-KE elevates Qwen2.5-7B's multi-hop QA accuracy to 90.2% (↑17.6 pp) while suffering merely 6.3% drop under heavy distraction and <1% when answers are leaked. Our quantitative analysis confirms Reason-KE's resilience and efficiency, establishing a new state of the art for reliable LLM knowledge updates. The code will be released.

1 Introduction

Large language models (LLMs, Grattafiori et al., 2024; Yang et al., 2024; Guo et al., 2025) have recently advanced at an unprecedented pace, demonstrating strong performance across a wide range of natural language processing tasks (Zhao et al., 2023). However, these models encode world knowledge statistically, and updating emerging facts via full retraining is prohibitively expensive. To address this limitation, knowledge editing (KE) techniques (Yao et al., 2023) have been proposed to inject or overwrite the specific facts in pretrained LLMs without retraining from scratch.

Existing KE methods fall into two main categories. Parameter modification approaches (Zhu et al., 2020; Meng et al., 2022a,b) directly alter model weights to integrate new information, while parameter preservation methods (Wang et al., 2024a; Cohen et al., 2024a) add lightweight modules or leverage in-context learning to achieve editing with minimal changes to the base model. Although parameter-preservation frameworks perform well on multi-hop question answering (MQA) benchmarks (Zhong et al., 2023; Cohen et al., 2024b), they often rely too heavily on surface-level context cues. As a result, their performance degrades sharply when faced with noisy or irrelevant facts, a scenario common in real-world applications (Wang et al., 2024b).

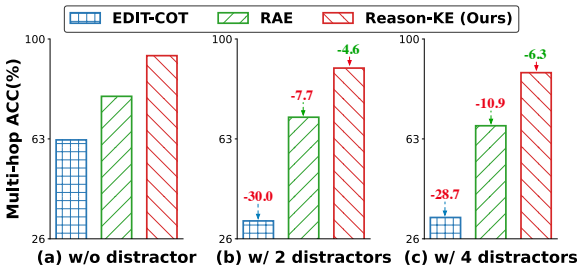

Figure 1: **Performance of three knowledge-editing methods under increasing levels of distraction**. (a) w/o distractor: Editing sets contain only relevant facts. (b) w/ 2 distractors: Two irrelevant facts are added for each relevant fact. (c) w/ 4 distractors: Four irrelevant facts are added for each relevant fact. Numeric annotations above each bar indicate the relative accuracy drop (↓) from the no-distractor setting. Our method consistently achieves the highest accuracy and exhibits the smallest degradation as the number of distractors grows.

To overcome these challenges, we introduce **Reason-KE**, an end-to-end reasoning-chain-based KE framework that guides a pretrained LLM through four structured stages: 1) *Acknowledgment* of updated information; 2) *Relevance Determination* to filter distractors; 3) *Selective Application* of pertinent facts; and 4) Final *Reasoning* to derive the answer. By explicitly modeling each reasoning step in a single pass, Reason-KE eliminates the need for complex iterative pipelines and maintains robustness under heavy distraction (Figure 1).

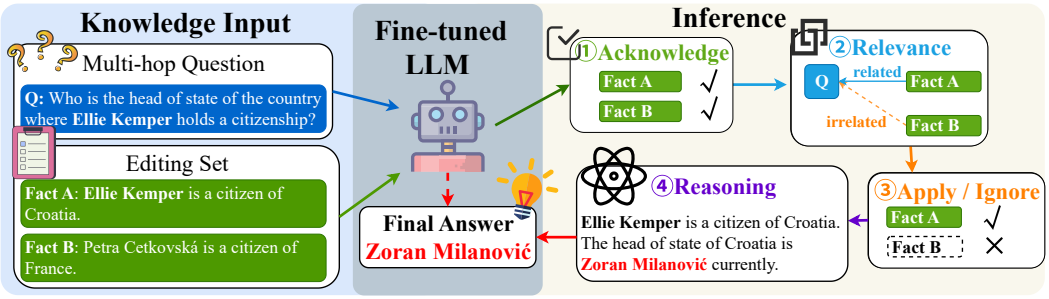

Figure 2: **Illustration of our four-stage workflow for multi-hop KE in the presence of distractors.** 1. *Acknowledge Updated Information* ingests all editing facts ("Fact A & B"). 2. *Determine Relevance* evaluates each fact against the question to gauge usefulness. 3. *Apply or Ignore* retains relevant facts ("Fact A"), discards others ("Fact B"). 4. *Reasoning* composes an explicit reasoning chain to derive the final answer ("Zoran Milanović"). This pipeline filters out noise and enables reliable multi-hop reasoning even in the presence of redundant information.

Experiments on the MQuAKE-CF dataset-augmented with up to four irrelevant facts-show that Reason-KE raises Qwen2.5-7B's multi-hop QA accuracy to 90.2%, a 17.6 percentage-point gain over the strongest baseline, while limiting performance drops to 6.3% under high distraction and below 1% when answers are directly exposed. Our **contributions** are threefold:

- We propose Reason-KE, a simple and effective end-to-end framework for multi-hop knowledge editing that robustly handles redundant information without iterative loops.
- We empirically demonstrate that our method consistently outperforms state-of-the-art (SOTA) baselines across diverse distractor settings and exhibits minimal degradation when answers are leaked.
- We provide quantitative analyses on the impact of distractors and answer exposure, establishing a new SOTA in both reliability and efficiency for LLM knowledge updates.

2 Methodology

2.1 Background

Knowledge Editing. Knowledge editing (KE) seeks to update specific facts in a pretrained LLM without full retraining (Mitchell et al., 2021). Each fact is a triplet $f = (s, r, o)$, and an edit changes the object to o^*, yielding $e = (s, r, o \rightarrow o^*)$, for example, (*the United States, the president of { } is, Joe Biden* $\rightarrow$ *Donald Trump*).

Multi-hop QA in KE. Multi-hop questions Q require reasoning over a chain of interdependent facts $C = [(s_1, r_1, o_1), \ldots, (s_n, r_n, o_n)]$, where $s_{i+1} = o_i$ and o_n is the answer. Under editing, any $e \in \mathcal{E}$ can alter the final result. Prior work finds the golden chain C^* via iterative decomposition (Wang et al., 2024a; Zhong et al., 2023; Gu et al., 2023), but demands highly relevant facts. In realistic settings, editing sets often include distractors, which may confuse the model, underscoring the need for noise resistance.

2.2 Reasoning Training

To mitigate the issue above, We propose Reason-KE, a simple yet efficient end-to-end framework. Unlike previous methods, Reason-KE eliminates complex iterative framework and instead explores the model's intrinsic reasoning ability. Specifically, given an editing set $\mathcal{E}$ and a corresponding question Q, the LLM is required to generate an explicit reasoning process that leads to the final answer.

Construction of Training Data. We used the widely used dataset MQuAKE-CF in the field of knowledge editing, which contains 9,218 data points, to construct our training dataset. Specifically, given an editing set $\mathcal{E}$ and its corresponding QA pair, we employ Deepseed-R1[1] to generate a step-by-step reasoning process. Moreover, to enhance the model's robustness to distractors, we add extra 0, 2, and 4 distractors to editing sets $\mathcal{E}$ at a ratio of 90%, 5% and 5%, respectively. To achieve our goal, our designed reasoning process includes the following parts: (1) **Acknowledge Updated Information:** Confirm the facts in $\mathcal{E}$. (2) **Determine Relevance:** Determine the relevance between the facts and question Q. (3) **Apply Updated Information or Ignore:** Based on the relevance in (2), determine which fact to apply or ignore. (4)

[1] https://api-docs.deepseek.com/

Reasoning: Based on the applied knowledge and Q, reason out the final answer. Further details are provided in Appendix A.

Fine-Tuning. We trained on Qwen2.5-7B-Instruct (Yang et al., 2024), and more training details can be found in Appendix B.3.

3 Experimental Setup

Baselines and Models. We compare our method with various model editing methods, including the parameter modification method ROME (Meng et al., 2022a) and parameter preservation methods MeLLo (Zhong et al., 2023), PokeMQA (Gu et al., 2023), EditCoT (Wang et al., 2024a), and RAE (Shi et al., 2024). All our experiments are conducted on Qwen2.5-instruct-7B (Yang et al., 2024). More details in Appendix B.1.

Datasets and Metrics. We evaluate our method and baselines on MQuAKE dataset (Zhong et al., 2023), a knowledge editing benchmark designed for multi-hop QA. We use MQuAKE-CF-3k as our test set, which includes 3,000 items. Importantly, MQuAKE-CF-3k and MQuAKE-CF share no overlapping data points. Following previous work (Zhong et al., 2023), we choose *Multihop-Accuracy* as evaluation metric, which use Exact Match to measure the accuracy. More details can be found in Appendix B.2.

Distractors Selection. To systematically evaluate robustness to irrelevant-fact interference, we retrieve k extra facts for each supporting fact needed by the question, where $k \in \{0, 1, 2\}$. Consequently, if a question requires m facts, the total number of distractors added to $\mathcal{E}$ is $n = m \times k$. See Appendix B.4 for more details.

4 Main Rsults

We demonstrate the result of our method and baselines in Table 1& 2. We demonstrate that:

Reason-KE outperforms other methods in multi-hop setting by a significant margin. As shown in Table 1, our method achieves a 17.64% average performance improvement over RAE in multi-hop settings, demonstrating strong reasoning capability. Moreover, when encountering distractors, prior methods like PokeMQA and EDIT-COT struggle to construct the golden path. In contrast, our method enables the model to reason internally, preserving consistency throughout the reasoning chain. Even under the most challenging 4-hops setting, our method still maintains high performance.

Reason-KE brings stable improvements on more complex editing sets $\mathcal{E}$. The more facts that need editing, the more interference terms added and the more complex the editing set $\mathcal{E}$ becomes. This setting demands that the model not only comprehends the facts in $\mathcal{E}$ but also exhibits robust reasoning capabilities to arrive at the final answer. As demonstrated in Table 2, most existing methods exhibit significant performance degradation in multi-edit scenarios (number of edits > 1). Although RAE employs pruning strategies to filter some redundant information, it does not enhance the model's fundamental capability, resulting in suboptimal performance. In contrast, our method can independently identify which facts are relevant to the question. This enables it to effectively filter out distractors even in complex multi-edit conditions (number of edits > 2), achieving the best performance.

Reason-KE does not rely on the leakage of answers in facts. In our analysis of the MQUAKE-CF-3K dataset, we found that in 1,852 instances, the object o^* of the fact triple (s, r, o^*) aligns exactly with the final answer o_n^* of the multi-hop question Q. This observation raises concerns about potential shortcut learning, whereby models may obtain answers directly from these fact triples without truly comprehending the updated information. To investigate, we isolate instances where answers are directly exposed and conduct experiments. As Table 3 shows, when introducing distractors to $\mathcal{E}$ under leakage conditions, most methods' performance drops sharply, revealing their dependence on superficial pattern matching. However, our method exhibits a performance drop of less than 1%, proving its independence from surface-level editing.

5 Analysis

Does every composition of the training data matter? We focus on the importance of the reasoning process in Reason-KE. As shown in Table 4, each component contributes to the reasoning capabilities of model. Removing any particular segment of the training data disrupts the chain of reasoning and consistently degrades performance. Notably, completely removing the reasoning process induces over-dependence on the updated facts, leading to a sharp performance decline when facing distractors, with an average performance drop of 29.89%.

Method	2-hops			3-hops			4-hops			*Avg.*
	w/o Distr.	w/ 2 Distr.	w/ 4 Distr.	w/o Distr.	w/ 2 Distr.	w/ 4 Distr.	w/o Distr.	w/ 2 Distr.	w/ 4 Distr.	
ROME	12.00	12.00	11.99↓	8.83	8.95	9.11	5.46	5.68	5.50	8.84
Mello	80.90	70.90↓	65.80↓↓	40.30	29.50↓	30.40↓↓	9.30	10.40	11.00	38.72
PokeMQA	84.10	77.80↓	78.30↓	61.40	50.90↓	49.40↓↓	16.00	12.70↓	9.10↓	48.86
EditCoT	76.40	51.80↓↓	54.70↓↓	44.00	16.10↓↓	16.90↓↓	67.50	30.00↓↓	30.10↓↓	43.06
RAE	88.90	87.50↓	85.30↓	71.10	60.10↓	58.10↓↓	76.30	65.50↓	60.20↓↓	72.56
Reason-KE	**97.00**	**96.70**↓	**96.70**↓	**88.90**	**85.20**↓	**84.80**↓	**95.60**	**85.80**↓	**81.10**↓	**90.20**

Table 1: **Multi-hop performance** with **bolded** best results. Baseline performance (0 distractors) is underlined in **w/o Distr.** columns, with interference impacts quantified in **w/ 2/4 Distr.** columns. ↓ indicates >6% performance drop from w/o Distr., ↓↓ denotes >12% catastrophic drop, and ↓ shows stable performance (<6%).

Method	#Edits: 1			#Edits: 2			#Edits: 3 & 4			*Avg.*
	w/o Distr.	w/ 2 Distr.	w/ 4 Distr.	w/o Distr.	w/ 2 Distr.	w/ 4 Distr.	w/o Distr.	w/ 2 Distr.	w/ 4 Distr.	
ROME	9.36	9.37	9.47	9.81	9.97	9.97	6.66	6.85	6.70	8.68
Mello	41.54	35.41↓	32.11↓	55.67	48.83↓	47.70↓	30.60	23.81↓	25.24↓	37.88
PokeMQA	59.38	54.35↓	53.34↓	63.92	56.23↓	55.48↓	33.81	26.19↓	22.98↓	47.30
EditCoT	64.59	49.86↓↓	49.13↓↓	64.57	35.52↓↓	39.46↓↓	57.62	6.55↓↓	7.02↓↓	41.59
RAE	65.97	63.04↓	60.11↓	81.07	68.98↓↓	67.39↓↓	92.50	84.05↓	78.57↓↓	73.52
Reason-KE	**89.84**	**84.08**↓	**84.26**↓	**97.00**	**90.25**↓	**85.85**↓	**95.00**	**94.64**↓	**93.93** ↓	**90.54**

Table 2: **Multi-edit performance** with best results **bolded**. All markers follow the same conventions as Table 1.

Method	Answer w/ exposed		
	w/o Distr.	w/ 2 Distr.	w/ 4 Distr.
Mello	56.75	48.06 (↓8.69)	46.54 (↓10.2)
PokeMQA	60.21	51.30 (↓8.91)	50.27 (↓9.94)
EditCoT	64.25	25.49 (↓38.8)	27.32 (↓36.9)
RAE	94.98	88.82 (↓6.16)	85.15 (↓9.83)
Reason-KE	**97.08**	**96.70** (↓0.38)	**96.71** (↓0.37)

Table 3: **Performance in the answer-exposed setting**, where ↓ indicate >5% degradation from w/o Distr. and ↓ arrows denote <1%.

Method	Multi-hop acc			*Avg.*
	w/o Distr.	w/ 2 Distr.	w/ 4 Distr.	
Reason-KE	**93.83**	**89.23**	**87.53**	**90.20**
-w/o acknowledge	92.77↓	81.00↓	78.67↓	84.15↓
-w/o relevance	88.76↓	86.47↓	80.40↓	87.18↓
-w/o apply	87.77↓	74.70↓↓	73.67↓↓	78.71↓
-w/o reasoning	92.17↓	81.77↓	78.43↓	84.12↓
-w/o Distr. sample	93.47↓	88.40↓	86.17↓	89.35↓
-only answer	91.97↓	50.60↓↓	38.37↓↓	60.31↓↓

Table 4: **Ablation study results under varying distractor settings**, showcasing the impact of different components on the multi-hop accuracy. ↓ and ↓ are the same as Table 1.

Similarly, excluding distractor-handling samples from the training data also impairs the model's anti-interference ability.

Reason-KE demonstrates high efficiency in various scenarios. We investigate the inference efficiency of Reason-KE and other in-context-learning methods using 100 randomly sampled questions from MQuAKE-CF-3k. Figure 3 illustrates the average time each method needs to edit n cases

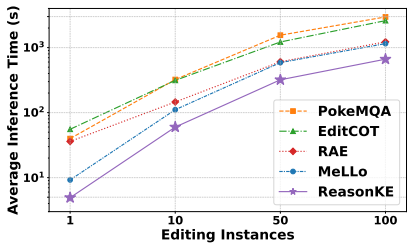

Figure 3: **Average inference time** for n editing instances, where $n = 1, 10, 50, 100$.

($n = 1, 10, 50, 100$), when facing k distractors ($k \in \{0, 1, 2\}$.). Due to their high dependence on iterative strategies, most existing methods require more time to reach a final answer, especially in complex scenarios. In contrast, Reason-KE only requires an editing prompt to perform reasoning, achieving a significant efficiency improvement.

6 Conclusion

We present Reason-KE, a streamlined end-to-end framework that guides an LLM through four explicit reasoning stages—acknowledgment, relevance filtering, selective application, and final inference—to perform multi-hop knowledge edits in one pass. On MQuAKE-CF-3K, Reason-KE boosts Qwen2.5-7B's accuracy to 90.2% (↑17.6 pp) while limiting drops to 6.3% with four distractors and under 1% when answers leak. Ablations confirm that each reasoning stage is critical to robustness, and inference is markedly faster than iterative baselines. Reason-KE offers a simple, reliable, and efficient solution for updating LLM world knowledge.

Limitations

While our work presents promising results, several limitations should be noted. First, due to computational constraints, we validate Reason-KE on models with up to 7B parameters. Evaluating larger models, especially those exceeding 70B parameters, could provide more comprehensive insights. Second, although our method performs well in multi-hop settings, its potential in other domains like finance or law remains unexplored. Future research will focus on scaling Reason-KE and extending its application to additional fields beyond the knowledge triple setup.

Ethics and Reproducibility Statements

Ethics We take ethical considerations seriously and strictly adhere to the ACL Ethics Policy. All datasets used in this work are publicly available and widely adopted by the research community. Our methods focus on enhancing the multi-hop QA knowledge editing capabilities of large language models without introducing harmful biases or unethical content. We ensure that all experiments are conducted in compliance with ethical guidelines, emphasizing fairness and transparency in model deployment.

Reproducibility In this paper, we discuss the detailed experimental setup, including training hyperparameters, baseline implementations, and statistical descriptions. More importantly, ***we have provided our code and data in the Supplementary Material*** to help reproduce the experimental results of this paper.

References

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024a. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024b. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hengrui Gu, Kaixiong Zhou, Xiaotian Han, Ninghao Liu, Ruobing Wang, and Xin Wang. 2023. Pokemqa: Programmable knowledge editing for multi-hop question answering. *arXiv preprint arXiv:2312.15194*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

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 *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2056–2066.

Changyue Wang, Weihang Su, Qingyao Ai, and Yiqun Liu. 2024a. Knowledge editing through chain-of-thought. *arXiv preprint arXiv:2412.17727*.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, and 1 others. 2023. Easyedit: An easy-to-use knowledge editing framework for large language models. *arXiv preprint arXiv:2308.07269*.

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

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

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. *arXiv preprint arXiv:2305.13172*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv 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. *arXiv preprint arXiv:2012.00363*.

A Details of Dataset Construction

For each item in MQuake-CF, we generate a corresponding reasoning process. In total, we construct 9,218 data entries for the training set. Here, we present the detailed prompts used for sample generation. Specifically, we employ the following prompts to guide LLMs to produce the reasoning process.

Reasoning Process Generation Prompt

Please provide a reasoning process based on my following tasks and corresponding answers. Your answer must strictly follow the steps of my example.

[Task]:Please acknowledge the updated information provided below and respond to the subsequent query.
[Updated Information]:Roblin Park is located in New South Wales.
[Query]:What is the capital city of the state where Roblin Park is located?
[Answer]:Sydney
[Reasoning Process]
1.Acknowledge Updated Information: The updated information states that Roblin Park is located in New South Wales.
2.Determine Relevance: The query asks for the capital of the state where Roblin Park is located. Since the updated information explicitly provides the state (New South Wales), it is directly relevant to answering the question.
3.Apply Updated Information or Ignore: Apply Roblin park's new location.
4.Reasoning: Roblin Park lies within the state of New South Wales. The capital of New South Wales is Sydney. Therefore, the capital of the state where Roblin Park is located is Sydney
[Answer]: Sydney

[Task]:Please acknowledge the updated information provided below and respond to the subsequent query.
[Updated Information]: `<updated_information>`
[Query]: `<query>`
[Answer]: `<answer>`

B Details of Experimental Setup

B.1 Details of Baselines

We compare Reason-KE with parameter modification method and current In-Context Editing methods:

ROME (Meng et al., 2022a) employs causal mediation analysis to pinpoint the target area for editing and subsequently updates the parameters of the feed-forward network. In our implementation, we utilize EasyEdit (Wang et al., 2023) with its default settings.

MeLLo (Zhong et al., 2023) employs the plan-and-solve strategy to perform in-context editing. It first decomposes the problem into sub-questions and uses retrieval. Following the official setting, the prompts were adapted to Instruct Models, and the maximum number of retrieval rounds was fixed at four.

PokeMQA(Gu et al., 2023) building upon Mello, enhances question understanding by prompting large language models (LLMs) to decompose knowledge-augmented multi-hop questions. We adhere to the official settings, allowing up to five rounds of interaction and utilizing their pre-trained Scope-Detector.

EditCoT (Wang et al., 2024a) employs an iterative CoT approach. It first generates an initial CoT based on user input and the direct answer. A dedicated CoT editor then revises this reasoning trace, injecting newly retrieved knowledge to resolve any conflicts or gaps. Once the CoT has been updated, the language model is prompted to reason along the refined path and generate the final answer. Following the official setting, the maximum number of retrieval rounds was fixed at four.

RAE (Shi et al., 2024) constructs retrieval-oriented knowledge graphs and uses the model to optimize graph retrieval and pruning.

B.2 Details of Datasets

Table 5 shows the statistics of the MQUAKE-CF-3k datasets, which contain 3000 data points.

Datasets	#Edits	2-hop	3-hop	4-hop	Total
MQUAKE-CF-3K	1	513	356	224	1093
	2	487	334	246	1067
	3	-	310	262	572
	4	-	-	268	268
	All	1000	1000	1000	3000

Table 5: Statistics of MQuAKE-CF-3K datasets.

B.3 Implementation Details

All experiments were executed on 4 NVIDIA A100 GPUs (80G). We implement our Reason-KE method by supervised fine-tuning on Qwen2.5-7B-Instruct (Yang et al., 2024), with the training duration approximately lasting 100 to 120 minutes. Hyperparameters of our Reason-KE method are in Table 6.

Hyperparameter	Standard-FT
Epoches	5
Warmup ratio	0.05
Optimizer	AdamW
Learning rate	1e-5
Scheduler	cosine
Weight decay	1e-4
Block size	32768

Table 6: **Hyper-parameters** for training our Reason-KE.

B.4 Details of Distractors Selection.

We employ Contriever (Izacard et al., 2021) for retrieval and TopK (Liu et al., 2021) as the baseline selection method. Specifically, for each edit fact, we retrieve from the pre-edit corpus and match with the Top-k corresponding post-edit facts, where $k \in 0, 1, 2$.

C Used Scientific Artifacts

Below is a list of the scientific artifacts used in our work. Our use of these existing artifacts is consistent with their intended use.

- *DeepSpeed (Apache-2.0 license)* [2], a deep learning optimization library that enhances the efficiency of large language model training.
- *Transformers (Apache-2.0 license)* [3], a framework offering state-of-the-art pretrained models for NLP tasks.
- *trl (Apache-2.0 license)* [4], a library that combines reinforcement learning (RL) with transformer models.

[2] https://github.com/deepspeedai/DeepSpeed
[3] https://github.com/huggingface/transformers
[4] https://github.com/huggingface/trl