Reason-KE++: Aligning the Process, Not Just the Outcome, for Faithful LLM Knowledge Editing

Anonymous ACL submission

Abstract

Aligning Large Language Models (LLMs) to be faithful to new knowledge in complex, multi-hop reasoning tasks is a critical, yet unsolved, challenge. We find that SFT-based methods, e.g., Reason-KE (Wu et al., 2025b), while state-of-the-art, suffer from a "faithfulness gap": they optimize for format mimicry rather than sound reasoning. This gap enables the LLM's powerful parametric priors to override new contextual facts, resulting in critical factual hallucinations (e.g., incorrectly reasoning "Houston" from "NASA" despite an explicit edit). To solve this core LLM alignment problem, we propose Reason-KE++, an SFT+RL framework that instills process-level faithfulness. Its core is a Stage-aware Reward mechanism that provides dense supervision for intermediate reasoning steps (e.g., Decomposition, Sub-answer Correctness). Crucially, we identify that naive outcome-only RL is a deceptive trap for LLM alignment: it collapses reasoning integrity (e.g., 19.00% Hop acc) while superficially boosting final accuracy. Our process-aware framework sets a new SOTA of 95.48% on MQUAKE-CF-3k (+5.28%), demonstrating that for complex tasks, aligning the reasoning process is essential for building trustworthy LLMs. Our code is available at: `https://anonymous.4open.science/r/test--1D47`.

1 Introduction

Large language models (LLMs) (Grattafiori et al., 2024; Yang et al., 2024; Guo et al., 2025a) have shown strong capabilities (Zhao et al., 2023), but their fixed parameters struggle with changing world knowledge. Consequently, knowledge editing (KE, Yao et al. (2023)) has emerged to enable precise modification of specific facts. Current methods are broadly categorized as parameter modification and parameter preservation.

Parameter modification methods (Zhu et al., 2020; Meng et al., 2022, 2023) can directly change

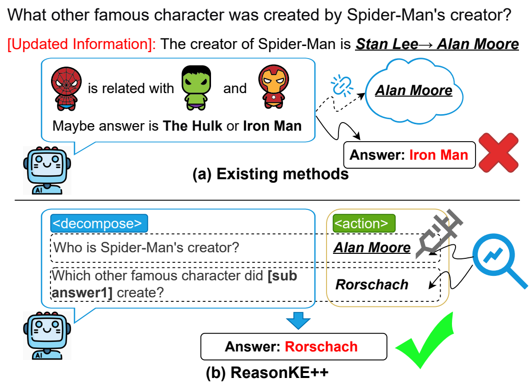

Figure 1: **An illustration of our core motivation**. (a) Existing methods often take an unfaithful shortcut based on strong priors, ignoring updated information and leading to incorrect answers. (b) Our ReasonKE++ decomposes the multi-hop query, ensuring a faithful reasoning process that correctly utilizes the new knowledge.

specific parameters to integrate new knowledge. However, current research (Zhang et al., 2024; Zhong et al., 2023) doubts whether they merely performs surface-level editing without truly understanding. In contrast, parameter preservation methods (Wang et al., 2025; Cohen et al., 2024a) achieve remarkable success by adding extra modules or leveraging the in-context-learning ability of LLMs. Currently, the parameter preservation KE framework performs well in multi-hop question answering (MQA) tasks (Zhong et al., 2023; Cohen et al., 2024b), which require models to reason based on updated information. Although in-context learning ability enhances models' understanding of updated knowledge (Zheng et al., 2023), it can also lead to excessive reliance on facts in the context. So when they encounter noisy or irrelevant knowledge, their performance drops sharply.

Furthermore, existing Knowledge Editing (KE) methods neglect the faithfulness of the reasoning process. We find that SFT-based methods, e.g., Reason-KE (Wu et al., 2025b), suffer from a critical "faithfulness gap": they optimize for format mimicry, enabling the LLM's powerful parametric priors to override new facts. This often leads to an unfaithful "shortcut" (see Figure 1a), where the model ignores the updated knowledge (e.g., "Alan Moore") and defaults to its pre-trained association (e.g., "Stan Lee $\rightarrow$ Iron Man").

To solve this, we propose **Reason-KE++**, which ensures a faithful reasoning process by enforcing a structured decomposition of the query (see Figure 1b). Reason-KE++ is a novel framework designed to fully unleash the model's multi-hop reasoning capabilities while maintaining robustness against distractors. It tackles problems through a meticulously designed reasoning process, which consists of three steps: 1) *Acknowledge* updated information and the question; 2) *Decompose* the question into sub-questions; and 3) *Act* by sequentially answering these sub-questions to derive the final solution. Inspired by Reason-KE, Reason-KE++ explicitly outputs these multiple reasoning steps within a single pass, which circumvents the reliance on complex iterative pipelines.

Specifically, our Reason-KE++ framework involves two phases: (1) The first stage focuses on Cold-Start Supervised Fine-Tuning (SFT) to instill initial reasoning patterns in the LLM. (2) The second stage transitions to reinforcement learning. **Crucially, we identify that naive, outcome-only RL is a deceptive trap**: our experiments (see Table 6) show it *collapses* reasoning integrity (e.g., 19.00% Hops acc) while superficially boosting final accuracy. To solve this, we introduce a novel **Stage-aware Reward mechanism**. Unlike sparse, outcome-only signals, our method employs a hierarchical reward structure that evaluates both the final answer's correctness and the validity of intermediate reasoning steps. This granular feedback loop discourages shortcut learning and ensures true process-level faithfulness.

We validated the effectiveness across various datasets across several models. Notably, on the MQuAKE-CF-3k dataset, Reason-KE++ achieved a multi-hop QA accuracy of 95.48%, marking a significant improvement of 5.28% over Reason-KE. Moreover, Reason-KE++ exhibits superior reasoning quality by generating more coherent reasoning paths and effectively preventing shortcut learning. The model also demonstrates strong robustness to severe distractions, with its performance declining by only 5.06% under such conditions.

Our **contributions** are threefold:

- We propose Reason-KE++, a novel two-stage (SFT+RL) framework for knowledge editing that employs a Stage-aware Reward mechanism. Our mechanism decomposes complex reasoning tasks into multiple assessable stages and provides step-by-step supervision, significantly improving the faithfulness of the model's reasoning.

- We demonstrate that Reason-KE++ substantially enhances model robustness and mitigates shortcut learning. By explicitly rewarding valid intermediate reasoning steps, our framework trains the model to construct coherent lines of reasoning and ignore irrelevant information, maintaining high performance even in the presence of severe distractors.

- We conduct comprehensive experiments on multiple knowledge editing benchmarks, where Reason-KE++ achieves state-of-the-art performance across diverse distractor settings.

2 Preliminary

2.1 Knowledge Editing of LLMs.

The goal of knowledge editing is to efficiently modify specific knowledge encoded within an LLM's parameters (Mitchell et al., 2022). A fact is represented as a triplet $f = (s, r, o)$, where s denotes the subject, r the relation, and o the object. The knowledge editing operation updates the object, expressed as $e = (s, r, o \rightarrow o^*)$, for example, *(the United States, the president of { } is, Joe Biden $\rightarrow$ Donald Trump)*. After editing, the model is expected to respond with the updated object "Donald Trump" to a relevant query (e.g., "Who is the president of the United States?").

2.2 Multi-hop QA within Knowledge Editing.

Unlike one-hop questions, answering a multi-hop question Q requires reasoning over a sequence of interdependent facts, or a "chain of facts," $C = [(s_1, r_1, o_1), ..., (s_n, r_n, o_n)]$, where $s_{i+1} = o_i$ and o_n is the final answer to Q. Under the knowledge editing setting, any alteration to this chain can change the final answer.

Specifically, given a base LLM p_θ and an editing set $\mathcal{E} = \{e_1, e_2, ..., e_m\}$, the task is to produce an edited LLM p_θ^* that can correctly answer a corresponding multi-hop question Q. Most previous works (Wang et al., 2025; Zhong et al., 2023; Gu et al., 2023) attempt this by finding the "golden path" $C^* =$

$[(s_1, r_1, o_1), ..., (s_i, r_i, o_i^*), ..., (s_n^*, r_n, o_n^*)]$ for Q through decomposition and iterative frameworks.

However, this approach faces two fundamental challenges. First, supervising the correctness of intermediate steps is difficult, failing to prevent **procedural shortcuts**. Second, these methods often overlook the **noise problem** endemic to real-world scenarios, where the editing set $\mathcal{E}$ may include redundant or irrelevant information. Addressing this gap requires a framework that can ensure step-by-step reasoning faithfulness while simultaneously mitigating the impact of distractors.

3 Methodology

Reason-KE++ is an RL-based framework. Unlike prior RL approaches, Reason-KE++ decomposes complex reasoning into multiple, evaluable stages. By designing a specific reward score for each stage, it transforms the final reward score from sparse to dense. This mechanism guides the model to construct faithful reasoning pathways and effectively mitigates shortcut learning. As shown in Figure 2, Reason-KE++ consists of two stages: a cold-start supervised fine-tuning phase to teach basic reasoning patterns, followed by a reinforcement learning phase with a Stage-aware Reward mechanism.

3.1 Reasoning Process Design

To ensure the model's reasoning is both transparent and faithful, we designed a structured process to guide its thinking. The entire thought process is contained within `<think>...</think>` tags and is organized into three distinct stages, each demarcated by its own special tokens (e.g., `<acknowledge>...</acknowledge>`). These three stages are: (1) **Acknowledge:** the model confirms the updated knowledge and its relevance to the input query. (2) **Decompose:** then it breaks the main problem into a series of actionable sub-questions. (3) **Act:** it methodically solves each sub-question, explicitly showing the derivation of each intermediate answer highlighted using the `\boxed{}`. Following this detailed thought process, the final answer is delivered, enclosed within `<answer>` and `</answer>` tags. This structured, machine-parsable format is a necessary prerequisite, as it enables the fine-grained evaluation required by our Stage-aware Reward mechanism in Section 3.4.

3.2 Cold Start for Foundational Reasoning

To prepare for the subsequent reinforcement learning phase, we start with Supervised Fine-Tuning (SFT) to equip the LLM with the foundational capability to generate a structured reasoning process. To achieve this, we curated a high-quality dataset.

Specifically, our data creation process begins by extracting multi-hop QA pairs from the COUNTERFACT (Wang et al., 2024). We then employ a structured prompt template to guide an advanced teacher model (e.g., GPT-4o-mini) to produce a step-by-step reasoning process for each pair. Moreover, to ensure quality, we apply a strict verification protocol that rectifies formatting errors and discards non-compliant samples. This ensures all outputs have syntactic consistency and structural integrity, making them atomically verifiable for the RL stage. Finally, this SFT process trains the model to acknowledge new facts, decompose multi-hop questions, and logically derive the final answer, establishing a solid foundation for the next stage. More details can be found in Appendix A.

3.3 Reason-KE++

3.3.1 Training Algorithm

We train Reason-KE++ by employing the Proximal Policy Optimization (PPO) algorithm. The policy π_θ is updated by optimizing the PPO objective:

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}[(q, a) \sim \mathcal{D}, o_{\leq t} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q)] \left\{ \min \left[r_t(\theta)\hat{A}_t, \ \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t \right] \right\}, \tag{1}$$

where q represents the query with updated information, a is the corresponding ground-truth answer, o denotes the sequence of generated tokens, $\hat{A}_t$ is the estimated advantage, and ε is the clipping hyperparameter. The objective leverages the principle of importance sampling via the probability ratio $r_t(\theta)$ between the current (π_θ) and the old policy ($\pi_{\theta_{\text{old}}}$):

$$r_t(\theta) = \frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_{\theta_{\text{old}}}(o_t \mid q, o_{<t})}, \tag{2}$$

3.4 Stage-aware Reinforcement Learning

To improve LLMs in complex multi-step reasoning tasks, we focus not only on the correctness of the final answer but also on optimizing the logical consistency and interpretability of the reasoning process itself. Traditional outcome-based rewards are often too sparse for multi-step reasoning, failing to provide effective guidance for the model's intermediate steps. For instance, a model might arrive at the correct answer through flawed reasoning (i.e., "lucky guess"), or an entire chain of reasoning could collapse due to a minor intermediate error. A

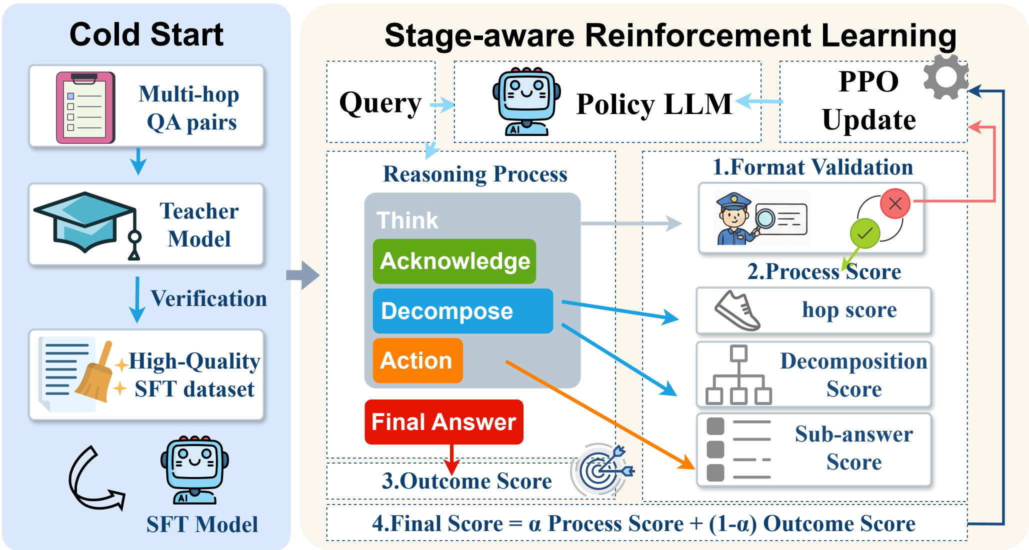

Figure 2: **The two-stage pipeline of ReasonKE++**. It starts with a cold-start SFT phase for foundational learning, followed by a Stage-aware Reinforcement Learning phase. In the RL stage, a dense reward signal, composed of a detailed Process Score and an Outcome Score, is used to optimize the model's ability to generate faithful reasoning.

sparse reward signal cannot effectively distinguish between these scenarios. To address this challenge, we introduce a Stage-aware Reward mechanism. The core is a structured reward function designed to perform a fine-grained evaluation and provide incentives for the model's performance at different stages of the reasoning process.

Format Validation We first rigorously check whether the model's output adheres to the pre-defined tag structure, ensuring that tags such as <think>, <decompose>, <action>, and <answer> are correctly paired and appear in the proper sequence. Furthermore, we validate the internal structure within the <action> tag, verifying that sub-questions and their answers conform to the [Sub question X] and `\boxed{}` formats. Any format violation results in a fixed penalty score (e.g., -1.0) and terminates further evaluation. This mechanism compels the model to learn to generate structured and parsable reasoning chains, which is fundamental for our fine-grained process assessment.

Once the output passes format validation, our reward function is composed of two primary components: a Process Score and an Outcome Score.

Process Score This is the cornerstone of our stage-aware method, designed to assess the quality of the model's reasoning process. It is further broken down into three sub-components:

(1) Hop Score: It assesses whether model has correctly identified the number of reasoning "hops" required to solve the problem. We score this by verifying if the number of generated sub-questions matches the number of predefined reasoning steps. A correct reasoning framework begins with an accurate assessment of the problem complexity.

(2) Decomposition Score: This component evaluates the quality of the sub-questions formulated by the model. We employ a pre-trained Sentence Transformer to convert both the model-generated sub-questions and the ground-truth sub-questions into vector representations. The cosine similarity between these vectors is then calculated to measure their semantic equivalence. High-quality decomposition is a prerequisite for reaching the correct solution, and this score incentivizes the model to learn how to break down complex problems into a series of logically coherent and solvable sub-tasks.

(3) Sub-answer Score: This part measures model's ability to correctly solve each sub-question. We individually check the correctness of the answer provided for each sub-question (enclosed `boxed{}`). The score is proportional to the ratio of correct sub-answers. This provides direct feedback for each intermediate step, encouraging the model to maintain high accuracy throughout the entire reasoning chain.

Outcome Score It evaluates the accuracy of the final answer. We extract the final answer generated by the model within the <answer> tags and compute F1 score against the ground truth. This ensures that the model's final output remains reliable.

Final Score If format validation fails, the final score is -1; otherwise, the final score is a weighted combination of the process score ($R_{process}$) and the outcome score ($R_{outcome}$):

$$R_{final} = \alpha \cdot R_{process} + (1 - \alpha) \cdot R_{outcome}, \quad (3)$$

where $\alpha \in [0, 1]$ is a hyperparameter that controls the trade-off between these two components.

Through this stage-aware reward mechanism, the training signal is transformed from sparse and monolithic to dense and multi-dimensional. It informs the model not only *if* it was correct, but *which* intermediate steps were flawed. This fine-grained feedback significantly improves the model's ability to learn complex reasoning strategies, leading to more logical, interpretable, and robust reasoning processes.

4 Experiments

4.1 Experimental Setup

Baselines and Models. We evaluate our approach against a parameter modification method (ROME (Meng et al., 2022)) and several parameter preservation methods (MeLLo (Zhong et al., 2023), PokeMQA (Gu et al., 2023), EditCoT (Wang et al., 2025), and RAE (Shi et al., 2024)). Most baselines and our method are implemented on Qwen2.5-instruct-7B (Yang et al., 2024). To demonstrate generalizability, we report performance on Llama3-8B-Instruct (Grattafiori et al., 2024). Further details for all baselines are in Appendix B.1.

Datasets and Metrics. Our evaluation leverages two distinct benchmarks: the MQuAKE dataset (Zhong et al., 2023), designed for multi-hop QA in knowledge editing, and the DUNE dataset (Akyürek et al., 2023), which focuses on generalized editing. For testing, we utilize the MQUAKE-CF-3k set (3,000 instances) and the Arithmetic, New-Info, and Scientific subsets from DUNE. For our RL training, we use the MQUAKE-CF set, which has no data overlap with our test set. Consistent with prior work (Zhong et al., 2023), we adopt *Multihop-Accuracy* as the primary metric. Further details are in Appendix B.2.

Distractors Selection. To systematically assess robustness, we introduce distractor facts into the evidence set $\mathcal{E}$. Specifically, for each of the m supporting facts required by a question, we add k distractors, where $k \in \{0, 1, 2\}$ represents the interference level. This amounts to a total of $n = m \times k$ distractor facts added per question. Further details are in Appendix B.4.

4.2 Main Results

The comparative performance is detailed in Tables 1 and 2. Our main findings are as follows:

Reason-KE++ consistently outperforms all other methods, especially in high-complexity scenarios. As shown in Tables 1 and 2, complex scenarios requiring deep reasoning (multi-hop) or dense information navigation (multi-edit) cause a sharp performance fall-off for most baselines. Although Reason-KE's explicit chains are strong, its SFT-based process is not fully optimized and shows performance degradation in demanding settings (e.g., 4-hop with distractors). Reason-KE++ addresses this by using reinforcement learning to optimize its structured reasoning template, achieving a more effective and faithful process. This yields substantial gains, outperforming Reason-KE by approximately 5% on average in both multi-hop and multi-edit settings and establishing new state-of-the-art results.

Reason-KE++ demonstrates superior robustness to irrelevant information. As shown in Tables 1 and 2, introducing distractor facts causes significant performance degradation for most baselines. Methods like MeLLo and EditCoT are particularly vulnerable, often experiencing catastrophic accuracy drops (marked by ↓↓) of over 12%. Even RAE, which employs filtering, suffers a noticeable decline. In stark contrast, both Reason-KE and Reason-KE++ maintain exceptionally stable performance (marked by ↓). This highlights that the explicit reasoning chain structure provides a strong defense, which our RL framework successfully maintains and enhances, effectively immunizing the model against noise.

Reason-KE++ Demonstrates Broad Generalizability and Superiority. To affirm our method's broad applicability, we evaluate it on a different LLM and a more diverse dataset. First, when deployed on Llama-3-8B-Instruct (Table 3), Reason-KE++ again establishes itself as the top-performing method, achieving a 2.6% average gain over Reason-KE, confirming its architectural advantages are model-agnostic. Furthermore, on the DUNE dataset (Table 4), Reason-KE++ delivers substantial improvements across all categories, especially on the complex "New-Info" (+7.1%) and

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	28.20	13.70↓↓	13.80↓↓	11.00	4.90↓	4.70↓	16.70	8.70↓	8.60↓	12.26
PokeMQA	82.20	52.80↓↓	51.00↓↓	46.90	20.80↓↓	18.90↓↓	56.10	18.80↓↓	19.40↓↓	40.77
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
Reason-KE++	**98.90**	**98.40**↓	**97.80**↓	**97.60**	**95.30**↓	**94.30**↓	**98.80**	**90.20**↓	**88.00**↓	**95.48**

Table 1: **Multi-hop QA performance** is shown, with the best scores in **bold**. We compare the baseline (underlined, **no distractors**) against performance with **2 or 4 distractors**. The resulting performance change is categorized as: stable (↓, <6% drop), significant (↓, >6% drop), or catastrophic (↓↓, >12% drop).

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	33.76	17.38↓↓	16.65↓↓	13.68	5.81↓	6.00↓	5.24	2.50↓	2.98↓	11.56
PokeMQA	57.27	27.26↓↓	28.18↓↓	68.70	38.05↓↓	37.21↓↓	58.69	26.19↓↓	22.38↓↓	40.44
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
Reason-KE++	**98.44**	**91.49**↓	**90.12**↓	**97.47**	**95.13**↓	**93.44**↓	**99.64**	**98.10**↓	**97.50**↓	**95.70**

Table 2: **Multi-edit performance** is presented, with the best results in **bold** and all markers retaining the same meaning as in Table 1.

Method	Multi-hop acc			*Avg.*
	w/o Distr.	w/ 2 Distr.	w/ 4 Distr.	
EditCoT	51.26	23.13↓↓	24.20↓↓	32.87
RAE	85.23	80.73↓	78.96↓	81.64
Reason-KE	94.37	89.47↓	87.53↓	90.46
Reason-KE++	**95.86**	**92.37**↓	**91.00**↓	**93.08**

Table 3: **Performance of `Llama-3-8B-Instruct` on MQuAKE-CF-3k**, presented using the same notational conventions as in Table 1.

Subset	Method	Acc			*Avg.*
		w/o Distr.	w/ 2 Distr.	w/ 4 Distr.	
Arithmetic	EditCoT	92.30	89.20↓	90.42↓	90.64
	Reason-KE	97.46	95.11↓	95.21↓	95.93
	Reason-KE++	**98.03**	**97.84**↓	**99.15**	**98.34**
New-Info	EditCoT	81.20	80.10↓	78.30↓	79.87
	Reason-KE	84.44	83.35↓	84.06↓	83.95
	Reason-KE++	**90.90**	**90.90**	**91.30**	**91.03**
Scientific	EditCoT	81.03	81.23↓	80.70↓	80.99
	Reason-KE	82.31	80.71↓	80.59↓	81.20
	Reason-KE++	**91.71**	**91.45**↓	**92.50**	**91.89**

Table 4: **Performance on the subsets of the DUNE dataset.** All symbols adhere to the same conventions as detailed in Table 1.

"Scientific" (+10.7%) subsets. This demonstrates that the enhanced reasoning process is highly effective for diverse, open-ended editing tasks.

5 Analysis

Reason-KE++ Consistently Adheres to The Gold Path. To verify that our model truly masters the capability of multi-hop reasoning rather than relying on heuristic shortcuts or stochastic coincidences, we conduct additional experiments using Hop-wise answering accuracy (Hop-wise acc). This metric rigorously assesses whether the model can correctly resolve each sub-question throughout the reasoning process. As illustrated in Table 5, while baseline methods such as PokeMQA and EditCoT suffer from a catastrophic performance collapse as the number of distractors increases, Reason-KE++ demonstrates exceptional robustness. Specifically, even under the most challenging setting with 4 distractors, Reason-KE++ maintains a high accuracy of 82.95%, representing only a marginal decay compared to the distractor-free environment. This demonstrates that Reason-KE++ consistently matches the gold reasoning path, ensuring that the final output is derived from a correct reasoning process rather than mere hallucination.

Method	Hop-wise acc			*Avg.*
	w/o Distr.	w/ 2 Distr.	w/ 4 Distr.	
PokeMQA	42.97	20.07 ↓↓	19.13 ↓↓	27.39
EditCoT	51.50	26.80 ↓↓	15.10 ↓↓	31.13
Reason-KE++	**92.90**	**85.51** ↓	**82.95** ↓	**87.12**

Table 5: **Performance using Hop-wise acc,** with the best results in **bold** and all markers retaining the same meaning as in Table 1.

Method	M-hop acc	Format acc	Hops acc	Sub acc	Similarity	Avg.
SFT	81.90	84.93	21.13	14.13	51.74	50.77
+ Outcome Score	94.72	73.50	19.00	16.33	54.73	51.66
++ Format Val.	95.56	99.77	39.83	29.67	55.80	64.13
+++ Hop Score	95.43	99.97	90.17	78.23	62.25	85.21
++++ Sub-ans Score	**95.64**	99.98	94.67	**87.30**	62.37	87.99
+++++ Dec. Score	95.48	**100.00**	**94.93**	87.17	**81.10**	**91.74**

Table 6: **Ablation study results with incremental components.** Cell colors indicate the performance gap compared to the SFT baseline, where blue denotes improvement and red denotes degradation. Best results in each column are **bolded**. Details of metrics can be found in Appendix B.5.

5.1 Ablation Study of Reason-KE++

To analyze the contribution of each component within our framework, we conducted a comprehensive ablation study. As presented in Table 6, a model trained via standard SFT acquires rudimentary reasoning skills but performs inadequately, highlighting the need for enhancement. We thus initiated reinforcement learning, starting with a naive, outcome-only reward ('+ Outcome Score'). While this approach yields a superficial boost in 'Multi-hop acc' (to 94.72), it proves to be a **deceptive trap**: the model fails to adhere to the desired format ('Format acc' drops to 73.50) and exhibits poor reasoning decomposition ('Hops acc' collapses to 19.00). This indicates the model has learned a new shortcut rather than a robust strategy.

To rectify this, we systematically integrated our process-oriented rewards. The introduction of each component demonstrably improves the quality of the reasoning process: '++ Format Validation' fixes structural integrity, '+++ Hop Score' dramatically enhances problem decomposition, and subsequent rewards refine the intermediate steps. This stepwise refinement culminates in our final 'Reason-KE++' model, which achieves holistic excellence across all metrics. This ablation validates that *supervising the entire reasoning process, not just the outcome, is indispensable for building a powerful and faithful reasoner*.

5.2 Case Study

We present a case study in Figure 3 to illustrate the fundamental difference in faithfulness between 'Reason-KE' (Wu et al., 2025b) and 'Reason-KE++'. When presented with a query and updated facts, both models correctly identify the relevant information (that NASA produced the Caddy) while discarding irrelevant distractors.

However, a critical divergence emerges in the subsequent reasoning. 'Reason-KE' defaults to its widely known (but in this context, incorrect) parametric prior, stating NASA's headquarters is in "Houston". This leads to an erroneous final answer, demonstrating a clear vulnerability to **factual hallucination**. In contrast, 'Reason-KE++' demonstrates superior reasoning capability. Its explicit '<decompose>' step breaks the complex query into simpler, verifiable sub-problems. This structured approach forces the model to perform more fine-grained reasoning, thereby correctly identifying "Washington D.C." as NASA's headquarters. This case illustrates that the *structured reasoning framework of 'Reason-KE++' is not merely a formatting preference but a crucial mechanism for ensuring step-by-step accuracy and mitigating error propagation, ultimately cultivating a more robust and trustworthy reasoner*.

6 Related Work

Knowledge Editing Knowledge Editing (Wang et al., 2024; Meng et al., 2022; Zhong et al., 2023; Wu et al., 2025a) efficiently updates LLM facts via parametric methods that modify weights (e.g., ROME (Meng et al., 2022), MEMIT (Meng et al., 2023)) or non-parametric In-Context Learning (ICL). Early ICL methods like MeLLo (Zhong et al., 2023), PokeMQA (Gu et al., 2023), and Edit-CoT (Wang et al., 2025) rely on ICL, which decomposes complex queries into simpler sub-tasks and applies fine-grained edits through carefully crafted prompts. More recently, Reason-KE (Wu et al., 2025b) departed from iterative ICL, employing Supervised Fine-Tuning (SFT) to generate an explicit, single-pass reasoning chain to solve the task. While this SFT approach established a new SOTA in robustness, it also introduced the "faithfulness gap" (see Figure 1) that our current work addresses.

Process-aware LLM Reinforcement Learning Reinforcement learning (RL) is widely used for

Question: What is the location of the headquarters of the manufacturer of Volkswagen Caddy?

Updated Information:

Fact 1 The company that produced Volkswagen Caddy is National Aeronautics and Space Administration.
Fact 2 The company that produced Ford Taunus V4 engine is SpaceX.
Fact 3 The company that produced Tu-2 is SpaceX.

Reason-KE (Erroneous Reasoning):

[**Reasoning Process**]

1. **Acknowledge Updated Information**: The updated information states that the Volkswagen Caddy is produced by the National Aeronautics and Space Administration (NASA).
2. **Determine Relevance**: The query asks for the headquarters location of the manufacturer of the Volkswagen Caddy. Since the updated information identifies NASA as the manufacturer, the headquarters of NASA is directly relevant.
3. **Apply Updated Information or Ignore**: Apply the updated manufacturer (NASA) for the Volkswagen Caddy.
4. **Reasoning**: The manufacturer of the Volkswagen Caddy is NASA. The headquarters of NASA is located in Houston, Texas. Therefore, the headquarters location is Houston.

[**Answer**]:
Houston

Reason-KE++: `<think>`

`<acknowledge>` The updated information states that the company that produced the Volkswagen Caddy is the National Aeronautics and Space Administration (NASA). The query asks for the location of the headquarters of the manufacturer of the Volkswagen Caddy.
`</acknowledge>`

`<decompose>` Break down the original problem into:
[Sub question 1] What company manufactured the Volkswagen Caddy?
[Sub question 2] What is the location of the headquarters of [sub answer 1]?
`</decompose>`

`<action>` Answer sub questions based on updated knowledge:
[Sub question 1]According to the updated information, the company that manufactured the Volkswagen Caddy is NASA, so the answer is National Aeronautics and Space Administration.
[Sub question 2] The headquarters of NASA is located in Washington D.C., so the answer is Washington D.C..
`</action>`
`</think>`

`<answer>`
Washington D.C.

Figure 3: **Case study comparing Reason-KE (Wu et al., 2025b) and Reason-KE++**. 'Reason-KE' defaults to its parametric prior (NASA → Houston), exhibiting factual hallucination. 'Reason-KE++' uses structured decomposition to faithfully arrive at the correct answer.

LLM alignment (e.g., RLHF (Ouyang et al., 2022), DPO (Rafailov et al., 2023)) and enhancing complex reasoning (e.g., DeepSeek-R1 (Guo et al., 2025b)), supported by efficient algorithms like GRPO (Shao et al., 2024).

However, the application of RL to the niche, challenging field of *multi-hop knowledge editing* remains largely unexplored. Our work addresses this critical gap. We demonstrate that naively applying RL with outcome-only rewards is a "deceptive trap" that collapses reasoning integrity (see Table 6). We are the first to show that to bridge the "faithfulness gap" left by SFT methods, a **Stage-aware Reward** mechanism is essential. Our approach pioneers the use of dense, process-level rewards to ensure LLMs can *faithfully* reason over new, edited knowledge in complex scenarios.

7 Conclusion

We identified a critical "faithfulness gap" in SFT-based methods for multi-hop knowledge editing (Wu et al., 2025b; Zhang et al., 2024). These methods optimize for format mimicry, enabling LLM priors to override new facts and cause factual hallucinations. To solve this, we proposed **Reason-KE++**, an SFT+RL framework that instills process-level faithfulness. Its core is a **Stage-aware Reward mechanism** that provides dense supervision for intermediate reasoning steps, such as decomposition and sub-answer correctness. Crucially, we found naive outcome-only RL is a "deceptive trap" that collapses reasoning integrity (e.g., 19.00% Hops acc). Our process-aware approach sets a new SOTA (95.48%), proving that for complex tasks, aligning the *process*, not just the *outcome*, is essential for building trustworthy LLMs.

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.

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

Afra Feyza Akyürek, Eric Pan, Garry Kuwanto, and Derry Wijaya. 2023. Dune: Dataset for unified editing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1847–1861.

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, 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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8069–8083.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, and 1 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, and 1 others. 2025b. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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*.

Yibin Lei, Liang Ding, Yu Cao, Changtong Zan, Andrew Yates, and Dacheng Tao. 2023. Unsupervised dense retrieval with relevance-aware contrastive pretraining. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10932–10940, Toronto, Canada. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. 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. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations 2023*.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2024. Revisiting demonstration selection strategies in in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9090–9101, Bangkok, Thailand. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

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. 2025. Knowledge editing through chain-of-thought. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, and 1 others. 2024. Easyedit: An easy-to-use knowledge editing framework for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*.

Yuchen Wu, Liang Ding, Li Shen, and Dacheng Tao. 2025a. Edit once, update everywhere: A simple framework for cross-lingual knowledge synchronization in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, Vienna, Austria. Association for Computational Linguistics.

Yuchen Wu, Liang Ding, Li Shen, and Dacheng Tao. 2025b. Robust knowledge editing via explicit reasoning chains for distractor-resilient multi-hop qa. In *Findings of the Association for Computational Linguistics: EMNLP 2025*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, 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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. 2024. Uncovering overfitting in large language model editing. *arXiv preprint arXiv:2410.07819*.

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*.

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 *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

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 PPO, we divided MQUAKE-CF into a training set and a test set, consisting of 8,296 and 922 samples, respectively. To construct the dataset for our cold-start phase, we began with the one-hop question-answering pairs from the COUNTERFACT dataset. For each pair, we generated a detailed, step-by-step reasoning process. Following a rigorous quality assurance protocol to filter out malformed or illogical samples, this procedure yielded a final, curated dataset of 718 instances. To ensure consistency and quality, we guided the teacher model using a carefully designed prompt template, which is detailed below.

Reasoning Process Generation Prompt

Please provide a reasoning process based on my following tasks and corresponding answers.

[Task]:Please acknowledge the updated information provided below and respond to the subsequent query.

[Updated Information]:
[Fact 1]Roblin Park is located in New South Wales.
[Fact 2]The Eiffel Tower is located in London.

[Query]:What is the capital city of the state where Roblin Park is located?

<think>
<acknowledge>The updated information states that Roblin Park is located in New South Wales. And the query is "what is the capital city of the state where Roblin Park is located?"</acknowledge>

<decompose>Break down the original problem into:
[Sub question 1]What state is Roblin Park located in?
[Sub question 2]What is the capital of [sub answer 1]? </decompose>

<action>Answer sub questions based on updated knowledge:
[Sub question 1]Detected relevant to [Fact 1], so the answer is `\boxed{New South Wales}`.
[Sub question 2]No relevant facts were detected, but [sub answer 1] can be applied, so the answer is `\boxed{Sydney}`.</action>
</think>

<answer>Sydney</answer>

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

B Details of Experimental Setup

B.1 Details of Baselines

We evaluate ReasonKE++ against two main categories of knowledge editing techniques: a parameter modification method and several in-context editing approaches.

ROME (Meng et al., 2022) leverages causal mediation analysis to precisely locate and modify specific weights within a model's feed-forward networks. This update directly overwrites the stored factual knowledge. For our experiments, we implement ROME using the EasyEdit library (Wang et al., 2024) with its default hyperparameter configuration.

MeLLo (Zhong et al., 2023) adopts a "plan-and-solve" methodology. It first deconstructs a complex query into simpler, solvable sub-questions, sequentially using retrieval to gather necessary information for each step. We follow the official implementation, adapting its prompts for instruction-tuned models and capping the retrieval process at four rounds.

PokeMQA (Gu et al., 2023) refines the initial question decomposition stage. It prompts the LLM to generate a better-structured reasoning plan after augmenting the query with relevant knowledge. Our setup mirrors the official configuration, which includes a maximum of five interaction rounds and the use of their provided pre-trained Scope-Detector.

EditCoT (Wang et al., 2025) focuses on iteratively refining a model's reasoning trace. It starts by generating an initial Chain-of-Thought (CoT) based on the query. A specialized editor module then revises this CoT, integrating retrieved knowledge to correct inconsistencies or fill informational gaps. The model is subsequently prompted to generate the final answer based on this refined reasoning path. In line with the original work, we limit the maximum number of retrieval rounds to four.

RAE (Shi et al., 2024) externalizes knowledge into a graph structure. It trains the model to perform optimized retrieval and pruning over this knowledge graph, effectively navigating the graph to find the correct information needed to answer the query.

Reason-KE (Wu et al., 2025b) was the first work to address the multi-hop knowledge editing problem by generating an explicit reasoning chain. However, Reason-KE was solely based on a standard Supervised Fine-Tuning (SFT) approach. While SFT can teach a model to mimic the format of a reasoning process, it lacks a dynamic mechanism to reward logical correctness or penalize unfaithful reasoning. This reliance on static examples makes the model unable to develop true robust reasoning capability.

B.2 Details of Datasets

Table 7 provides a statistical breakdown of the MQUAKE-CF-3k dataset, which consists of 3,000 instances.

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 7: Statistics of MQuAKE-CF-3K datasets.

B.3 Implementation Details

We implemented our Reason-KE++ framework by trained two recent large language models: Llama3-8B-Instruct (Grattafiori et al., 2024) and Qwen2.5-7B-Instruct (Yang et al., 2024). The training for each model followed our two-stage pipeline, encompassing both supervised fine-tuning (SFT) and reinforcement learning (RL). All experiments were conducted on a server equipped with 8 NVIDIA A100 (80GB) GPUs, and the entire training process required approximately 360 to 400 minutes per model. The specific hyperparameters used for training are detailed in Table 8.

B.4 Details of Distractors Selection.

We utilize Contriever (Izacard et al., 2021) to implement the TopK (Liu et al., 2022) retrieval-based baseline[1]. For each target fact requiring an edit, this method retrieves the top-k most similar post-edit examples from our dataset, where $k \in 0, 1, 2$.

[1]Note that better retrieval models, e.g., ReContriever (Lei et al., 2023), and exemplar selection methods (Peng et al., 2024) will improve the performance, but it is not the focus of this work.

Hyperparameter	SFT	RL
Learning rate (Actor)	1e-5	1e-6
Learning rate (Critic)	-	1e-5
Max sequence length	32768	1024
Batch size	1	2048
Optimizer	AdamW	AdamW
Scheduler	cosine	-
Weight decay	1e-4	-
Warmup ratio	0.05	-
KL coefficient	-	0.001
Training epochs	10	15

Table 8: **Hyper-parameters** for training Reason-KE++.

B.5 Details of Metrics.

Each metric is the average performance over settings with 0, 2, and 4 distractors. The metrics are defined as follows: **Multi-hop acc** is the accuracy of the final answer (EM); **Format acc** is the percentage of outputs adhering to the predefined format; **Hops acc** is the accuracy of the number of decomposed sub-questions matching the ground truth; **Sub acc** is the accuracy of intermediate `\boxed{}` answers; **Similarity** is the semantic similarity score between generated and ground truth sub-questions.

C Used Scientific Artifacts

Our work leverages several key open-source libraries to ensure reproducibility. We confirm that our use of these artifacts is in full compliance with their respective licenses and intended purposes.

- *DeepSpeed (Apache-2.0 license)* [2]: An optimization library used to enhance the efficiency and scale of large language model training.
- *Transformers (Apache-2.0 license)* [3],: The core framework providing the architectures and tools for the pre-trained language models used in NLP tasks.
- *trl (Apache-2.0 license)* [4]: A specialized library employed to implement the Supervised Fine-tuning and reinforcement learning phase.
- *verl (Apache-2.0 license)* [5]: A flexible, efficient, and production-ready RL training library for large language models (LLMs).

D Supplemental Experiment Results

D.1 Detailed Results of Answer Leakage

Our analysis revealed a potential data artifact in the MQUAKE-CF-3K dataset: in 1,852 instances, the object o^* of a supporting fact (s, r, o^*) directly coincides with the final multi-hop answer o_n^*. This 'answer leakage' raises a critical concern that models might learn a shortcut (i.e., answer extraction) rather than performing genuine reasoning. To test this, we created an 'answer-exposed' setting using only these instances and introduced distractors. As shown in Table 9, while most baselines suffer substantial degradation—with EditCoT plummeting by over 36%—revealing their heavy dependence on this shortcut, our methods demonstrate remarkable resilience. Both Reason-KE and Reason-KE++ maintain near-perfect stability, with performance drops of less than 1.5% (indicated by ↓). This confirms our framework promotes a true reasoning process, effectively ignoring superficial cues from answer leakage.

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

Method	Answer w/ exposed		
	w/o Distr.	w/ 2 Distr.	w/ 4 Distr.
Mello	11.34	4.59 (↓6.75)	5.56 (↓5.78)
PokeMQA	67.06	34.77 (↓32.29)	32.56 (↓34.5)
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)
Reason-KE++	**99.62**	**98.70**(↓0.92)	**98.27**(↓1.35)

Table 9: **Performance under the answer-exposed condition**, where ↓ marks a significant degradation (>5%) compared to the 'w/o Distr.' case, while ↓ denotes a stable performance with a minimal drop (<1.5%).

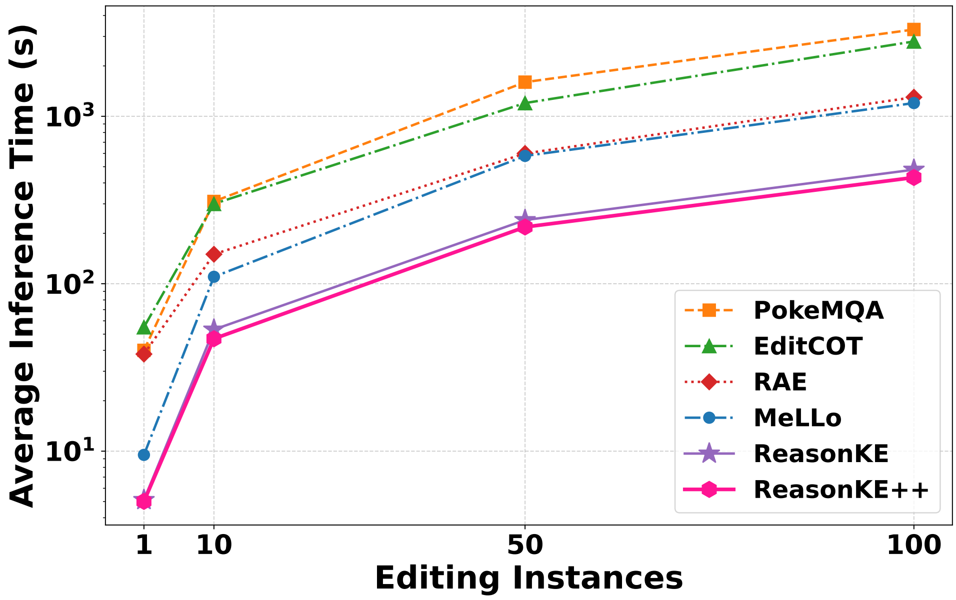

Figure 4: **Average inference time** for n editing instances, where $n = 1, 10, 50, 100$.

D.2 Detailed Results of Efficiency

Figure 4 illustrates the average inference time of Reason-KE++ compared to other methods. While Reason-KE already achieves significant efficiency gains by replacing iterative strategies with a simple reasoning prompt, Reason-KE++ further optimizes this performance. As shown in the figure, Reason-KE++ (the pink line) consistently maintains the lowest inference latency across all editing scales. This further improvement benefits from its superior reasoning accuracy. Following the evaluation protocol of MQuAKE-CF, each sample contains three multi-hop questions, and a sample is considered successful if any one of the questions is answered correctly. Thanks to its higher precision, Reason-KE++ is more likely to reach the correct answer in the initial attempts, thereby reducing the cumulative computational overhead per sample. Consequently, Reason-KE++ not only outperforms existing ICL methods but also slightly surpasses Reason-KE, achieving the best performance.