StruEdit: Structured Outputs Enable the Fast and Accurate Knowledge Editing for Large Language Models

Anonymous ACL submission

Abstract

As the modern tool of choice for question answering, large language models (LLMs) are expected to deliver answers with up-to-date knowledge. To achieve such ideal question-answering systems, locating and then editing outdated knowledge in the natural language outputs is a general target of popular knowledge editing methods. However, this target is challenging, as both identifying which tokens to edit in the reasoning steps and ensuring the coherence of the revised reasoning chain are difficult tasks. We argue that these challenges stem from the unstructured nature of natural language outputs. To address the above challenges, we propose **Structural Edit**ing (StruEdit), an improved baseline for knowledge editing. We first prompt LLMs to produce structured outputs consisting of reasoning triplets. Then, StruEdit removes any potentially outdated knowledge and efficiently refills the structured outputs with up-to-date information in a single step. Experimental results show that StruEdit consistently delivers the highest accuracy with lowest latency compared with other knowledge editing methods.

1 Introduction

With the widespread deployment of large language models (LLMs; OpenAI, 2022, 2023; Touvron et al., 2023a,b; Song et al., 2024), their reliability in answering questions is crucial, which entails accurately responding to queries with up-to-date knowledge. However, the knowledge used for pre-training LLMs cannot guarantee ongoing timeliness because the world is constantly changing. Knowledge editing (KE; Sinitsin et al., 2020; Zhu et al., 2020; De Cao et al., 2021) has been proposed to update the knowledge for LLMs.

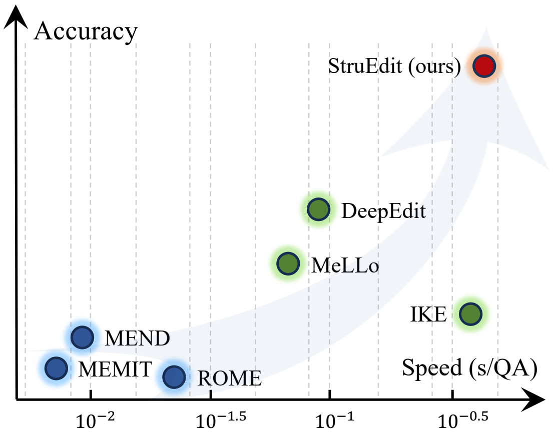


Figure 1: Comparison of performance between model editing (ME), in-context editing (ICE), and our StruEdit on multi-hop editing tasks, showing editing accuracy and average inference speed. Our StruEdit demonstrates the highest editing accuracy while maintaining the lowest latency.

The main process of existing KE methods can be summarized as *Locate-Then-Edit*, which requires accurately reflecting specific edited facts within the natural language reasoning steps. In a chain-of-thought (CoT) (Zhang et al., 2022) process, this means adjusting certain natural language reasoning steps based on new knowledge and accurately inferring the final result using that updated information. Model editing (ME; Meng et al., 2022a,b; Mitchell et al., 2022; Yao et al., 2023b; Xu et al., 2024) locates the position of knowledge to be edited, such as neurons in the FFN or matrix regions, and modifies them. In-context editing (ICE; Madaan et al., 2022; Zhong et al., 2023; Zheng et al., 2023; Wang et al., 2024; Bi et al., 2024a,d) locates relevant passages in the edit memory, prompting LLMs to utilize new knowledge to answer questions.

However, it is difficult to identify the tokens that need editing within the natural language reasoning steps, and incorrectly modifying parameters or providing inaccurate knowledge can directly result in editing failure. Additionally, editing the tokens while ensuring the coherence of the output reasoning chain is challenging, as conflicts between new knowledge and parametric knowledge (Petroni

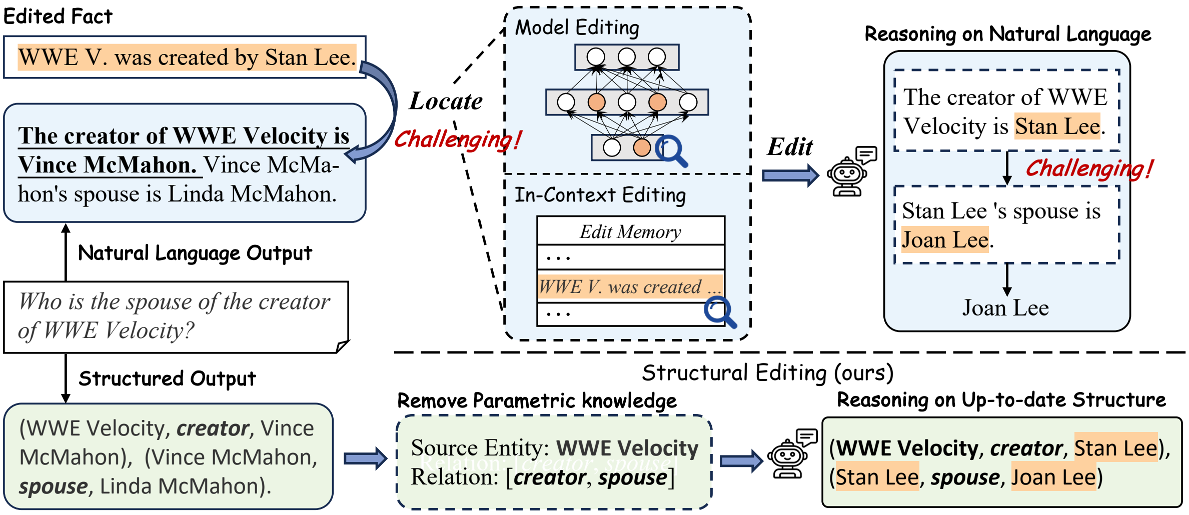

Edited Fact
WWE V. was created by Stan Lee.
The creator of WWE Velocity is Vince McMahon. Vince McMahon's spouse is Linda McMahon.
Natural Language Output
Who is the spouse of the creator of WWE Velocity?
Structured Output
(WWE Velocity, creator, Vince McMahon), (Vince McMahon, spouse, Linda McMahon).
Locate
Challenging!
Model Editing
In-Context Editing
Edit Memory
...
WWE V. was created ...
...
Edit
Reasoning on Natural Language
The creator of WWE Velocity is Stan Lee.
Challenging!
Stan Lee 's spouse is Joan Lee.
Joan Lee
Structural Editing (ours)
Remove Parametric knowledge
Source Entity: WWE Velocity
Relation: [creator, spouse]
Reasoning on Up-to-date Structure
(WWE Velocity, creator, Stan Lee),
(Stan Lee, spouse, Joan Lee)

Figure 2: Differences between ME, ICE methods, and our structural editing. ME and ICE first locate the position of edited facts within the natural language reasoning steps (ME identifies modification regions, while ICE retrieves relevant new knowledge) before editing. Both face challenges with incorrect localization and inconsistent reasoning due to the natural language output format. In contrast, structural editing removes LLMs' parametric knowledge and reasons over up-to-date knowledge structures using structured output logic to derive the final answer.

et al., 2020; Si et al., 2023; Xie et al., 2024) can lead to hallucinations during the reasoning process (Zhang et al., 2023; Huang et al., 2023a; Wang et al., 2023a) or make stubborn knowledge difficult to edit (Bi et al., 2024a).

In this paper, we argue that existing KE methods pose risks due to the *Locate-Then-Edit* approach based on natural language reasoning. We propose a new paradigm, structural editing, which structures the natural language outputs. Instead of relying on the two-step process of locating and editing, we directly remove all information potentially affected by new knowledge and refill the output based on the updated information. This approach eliminates the challenges caused by the coupling of different reasoning steps, enabling multi-step edits to be completed in a one-shot manner. Figure 2 shows the differences between structural editing and previous methods. To assess these approaches, we observe their performance on multi-hop editing tasks. We found that ME and ICE methods perform poorly when batch_size=full, with accuracy dropping significantly as the number of hops increases, indicating their difficulty in thoroughly editing knowledge. In contrast, the new structural editing demonstrates a high success rate and robustness, showcasing its potential.

Building on these insights, we propose an effective improved baseline for knowledge editing, called STRUEDIT. STRUEDIT edits LLM outputs through knowledge structures without the need to locate outdated knowledge and also the corresponding model parameters or input context. We use LLMs to refill new knowledge into the triplet reasoning structure based on specific logical rules, which accelerates reasoning speed and eliminates issues like hallucinations. Specifically, we extract the source entity and sequential relations from the reasoning chain, perform entity matching, and select relations in the knowledge structure to infer the reasoning path and obtain the final answer.

Experimental results demonstrate that our STRUEDIT consistently achieves the highest editing accuracy and the fastest speed compared to existing KE methods, as shown in Figure 1. STRUEDIT maintains robust editing capabilities as the number of reasoning hops and edited instances increases. Our work provides an improved KE baseline for LLMs with higher accuracy, faster performance, and greater robustness, paving the way for further advancements in KE.

2 Knowledge Editing on Multi-Hop Editing Tasks

In this paper, we focus on multi-hop editing tasks. Single-hop fact editing, such as modifying a fact triplet (s, r, t) to (s, r, t'), has been effectively addressed (Wang et al., 2023b). However, in real-world knowledge question answering (QA), changing one fact can trigger a "ripple effect" requiring updates to additional related facts (Cohen et al., 2024). Therefore, in multi-hop editing tasks, it is

Model	Method	batch_size=1				batch_size=full			
		2-hop	*3-hop*	*4-hop*	*avg.*	*2-hop*	*3-hop*	*4-hop*	*avg.*
LLaMA2-7B-Chat	ROME$^{\spadesuit}$ (Meng et al., 2022a)	35.4	20.3	16.2	23.9	4.2	2.5	0.7	2.5
	MEMIT$^{\spadesuit}$ (Meng et al., 2022b)	27.3	13.5	8.2	16.3	5.7	2.8	1.1	3.2
	IKE$^{\diamondsuit}$ (Zheng et al., 2023)	80.8	63.8	50.9	65.2	13.5	5.7	2.6	7.3
	MeLLo$^{\diamondsuit}$ (Zhong et al., 2023)	54.9	34.7	30.2	39.9	29.9	9.2	3.1	14.1
	Structural Editing (ours)	**100**	**100**	**100**	**100**	**91.5**	**90.7**	**56.8**	**79.1**
GPT-3.5-Turbo-Instruct	IKE$^{\diamondsuit}$ (Zheng et al., 2023)	78.5	76.2	73.4	76.0	17.3	9.6	6.7	11.2
	MeLLo$^{\diamondsuit}$ (Zhong et al., 2023)	72.6	48.7	40.5	53.9	47.8	20.2	16.8	28.3
	Structural Editing (ours)	**100**	**100**	**100**	**100**	**98.9**	**97.7**	**95.8**	**97.4**

Table 1: Experimental results (accuracy; %) on MQuAKE-2002 for multi-hop editing tasks (2, 3, 4-hop). We evaluated both open-source and closed-source LLMs across ME, ICE methods, and our StruEdit. Methods marked with $^{\spadesuit}$ belong to ME, while those marked with $^{\diamondsuit}$ belong to ICE. The best editing result on every LLM is highlighted in **bold** font.

crucial for LLMs to accurately reason the correct answer without introducing hallucinations caused by conflicts with parametric knowledge.

2.1 Multi-Hop Editing

Multi-hop editing is a more challenging task in KE, where LLMs need to consistently account for both the edited facts and related fact updates during multi-hop reasoning to ensure thorough knowledge revision. The main challenge lies in the potential conflict between new knowledge and the parametric knowledge in LLMs, which can result in factual hallucinations during reasoning (Bi et al., 2024b). For instance, regarding a two-hop fact chain (*WWE Velocity*, *created by*, *Vince McMahon*), (*Vince McMahon*, *spouse*, *Linda McMahon*). With a fact edit $(\textit{WWE Velocity}, \textit{created by}, \textit{Stan Lee})$ and an additional fact chain $(\textit{Stan Lee}, \textit{spouse}, \textit{Joan Lee})$, the correct updated answer should be *Joan Lee*.

2.2 Evaluation and Analysis

To thoroughly explore the editing capabilities of the main KE methods, including ME, ICE, and the new structural editing paradigm proposed in this paper, we conduct multi-hop editing experiments on the MQuAKE dataset (see Section 4.1 for details) with both open-source (LLaMA2-7B-Chat) and closed-source (GPT-3.5-Turbo-Instruct) models. Specifically, we construct multi-hop fact chains from the dataset and edit them with new knowledge based on each method. We set the batch size of the edit memory to 1 and full batch for KE evaluation. The batch size refers to the number of instances providing the edited facts for knowledge retrieval. A batch size of 1 means only the new knowledge relevant to the reasoning is provided, while a full batch simulates a real-world editing scenario where all new knowledge is provided, even if it is not directly related to the current reasoning task.

Table 1 presents the results of this experiment. From our observations, we found that structural editing consistently achieves the best performance. Notably, in the full batch size knowledge memory setting, ME and ICE methods perform poorly, while structural editing shows a significant lead on both open-source and closed-source models.

Furthermore, structural editing shows robust performance across varying batch sizes and reasoning hops compared to other methods. First, all methods show a noticeable drop in average accuracy when moving from batch_size = 1 to batch_size = full. In the ICE methods, IKE experiences the largest drop due to its struggle in retrieving effective new knowledge for complex reasoning chains, while MeLLo is less affected as it breaks down multi-hop queries into sequential single-hop queries. In the ME methods, additional parameter edits can lead to hallucinations. Our proposed structural editing method shows the smallest drop because, unlike other methods, it does not require locating specific knowledge, making it less affected by the number of editing instances.

As the number of reasoning hops increases, the accuracy of both ME and ICE methods decreases

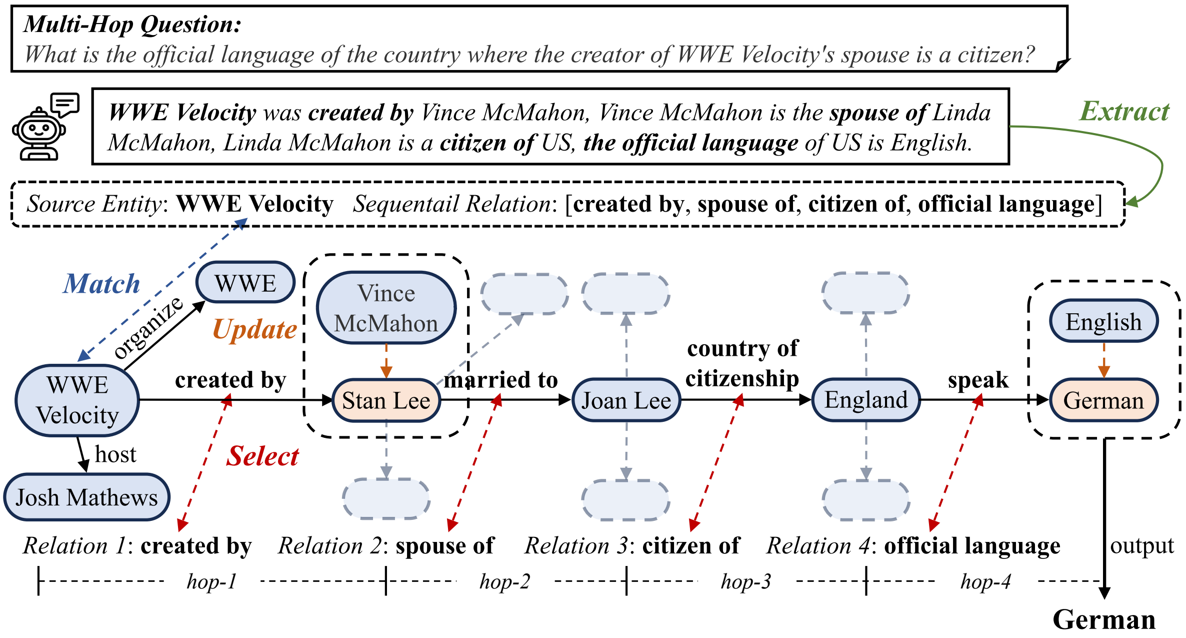

Multi-Hop Question:
What is the official language of the country where the creator of WWE Velocity's spouse is a citizen?
WWE Velocity was created by Vince McMahon, Vince McMahon is the spouse of Linda McMahon, Linda McMahon is a citizen of US, the official language of US is English.
Extract
Source Entity: WWE Velocity Sequentail Relation: [created by, spouse of, citizen of, official language]
Match
Update
Select
WWE
organize
Vince McMahon
WWE Velocity
created by
Stan Lee
married to
Joan Lee
country of citizenship
England
speak
English
German
host
Josh Mathews
Relation 1: created by
Relation 2: spouse of
Relation 3: citizen of
Relation 4: official language
output
hop-1
hop-2
hop-3
hop-4
German

Figure 3: An illustration showing how StruEdit answers multi-hop questions using new knowledge. For a multi-hop question, StruEdit first guides LLMs to generate a reasoning chain using their parametric knowledge. It then extracts the source entity and sequential relations, matches the source entity within an external knowledge structure, and selects based on the sequential relations during reasoning to arrive at the final answer.

to varying extents. Structural editing maintains 100% accuracy when batch_size=1 because, with the given structured fact chains and edited facts, LLMs can easily perform single-chain reasoning. Even with a full batch size, it remains stable, despite the need to consider more possible reasoning paths. This stability is due to the standardized knowledge representation in structured formats, which, compared to text, provides more reliable knowledge for LLMs' reasoning and reduces factual hallucinations.

3 StruEdit: An Improved Baseline of Knowledge Editing

Section 2 demonstrates the exceptional accuracy and robustness of structural editing in multi-hop editing tasks. To address more generalized multi-hop QA problems with new knowledge, we propose a more comprehensive method, StruEdit, an improved baseline for knowledge editing. The main idea behind ME and ICE methods is to combine edited facts with parametric knowledge, relying on the strong reasoning capabilities of LLMs to answer questions. However, this approach is implicit, as it's unclear whether the model's parameters were correctly updated or if the new knowledge is trusted by the LLMs. To reduce the burden on LLMs and the uncertainty of editing, StruEdit does not retain parametric knowledge and no longer targets specific knowledge for editing. Instead, all related knowledge is updated, allowing LLMs to reason over up-to-date knowledge structures based on the extracted logic of the question. Figure 3 illustrates the framework of StruEdit, using KGs as an example of the knowledge structure. We introduce the details of StruEdit from the following aspects.

3.1 Structrual Editing on Parametric Output

StruEdit uses up-to-date knowledge from the knowledge structure to edit LLMs' parametric output for multi-hop QA, leveraging the logic rules to reason over the structure. To enable reasoning over the knowledge structure for multi-hop questions, it is essential to provide the necessary conditions, including the source entity and sequential relations. First, we input the initial multi-hop question into the LLMs and use in-context demonstrations to guide them in generating a multi-hop reasoning chain. Then, we extract the source entity and sequential relations from this chain using LLMs, providing logic for subsequent reasoning.

In this process, we discard all other entities in the chain reflecting parametric knowledge without checking for conflicts with the new knowledge. We only utilize LLMs to obtain invariant relations, which are invariant over time in the reasoning chain, and entities are very unstable as connections

Entity / Relation Query Template

prefix_question: Which candidate entity/relation best matches the entity e_0 / relation r_i^t *<feature>*?

candidate_description: c_1: *<feature>*, c_2: *<feature>*, ..., $\mathrm{c}_{|C|}$: *<feature>*

Figure 4: The query template has two components: prefix_question, a selective question, and candidate_description, describing the candidate set $C = c_1, c_2, ..., c_{|C|}$, which represents either all entities or the relations associated with e_{i-1}. *<feature>* denotes the textual description of entities or relations.

therein. This reflects the core of our STRUEDIT, where we directly remove all information potentially affected by new knowledge and then refill it based on the updated knowledge. This ensures efficient and explicit editing.

3.2 Multi-hop Reasoning with LLMs

Multi-hop reasoning over a knowledge structure is key to our STRUEDIT approach. Formally, given a source entity e_0^t and a sequential relation $R = \{r_1^t, ..., r_h^t\}$ extracted according to **??** for an h-hop question, we ideally aim to find a reasoning path $P^t = \{(e_0^t, r_1^t, e_1), ..., (e_{h-1}, r_h^t, e_h)\}$ that leads to the final answer e_h.

However, although the source entity and sequential relations provide the reasoning logic, the accuracy of reasoning can be significantly impacted by discrepancies between the entities and relations extracted from the reasoning chain and those in the knowledge structure. For instance, as shown in Figure 3, "spouse of" in the sequential relation does not align with "married to" in the knowledge structure, which could lead to the selection of an alternate path during reasoning, ultimately resulting in an incorrect outcome. To address this issue, and inspired by Bi et al. (2024c), we adopt the following strategies when entities and relations cannot be precisely matched.

Entity Matching We construct a candidate set containing all entities, then query the LLMs with e_0^t to identify the most closely matching entity.

Relation Selection Similarly, during the i-hop reasoning, we construct a candidate set based on all relations $\{r_i^1, ..., r_i^m\}$ associated with the entity e_{i-1} to select the relation most similar to r_i^t.

The query template for entity matching and relation selection is shown in Figure 4. Entities and relations are aligned through LLM queries to optimally infer a reasoning path $P = \{(e_0, r_1, e_1), ..., (e_{h-1}, r_h, e_h)\}$ within the knowledge structure, leading to the final answer e_h, where e_0 best matches the extracted e_0^t and $\{r_1, ..., r_h\}$ most closely correspond to the extracted $\{r_1^t, ..., r_h^t\}$.

4 Experiments

4.1 Datasets and Tasks

Unlike the evaluation editing tasks in Section 2.2, we assess KE performance in the form of more generalized question answering tasks. We focus exclusively on the more realistic and challenging multi-hop tasks to assess whether the knowledge has been thoroughly edited. We conduct experiments using MQUAKE-3K (Zhong et al., 2023) along with its challenging derivatives, MQUAKE-2002 and MQUAKE-HARD, introduced by Wang et al. (2024). MQUAKE is a multi-hop QA benchmark for knowledge editing that provides multi-hop knowledge questions to evaluate KE on counterfactual edits. We construct KGs from the knowledge triples provided in MQUAKE to serve as the knowledge structure for our STRUEDIT.

4.2 Models and Baselines

We examine both closed-source models, including LLAMA2-7B-CHAT and LLAMA2-13B-CHAT, as well as open-source models, including GPT-3.5-TURBO-INSTRUCT and GPT-4O-MINI. We use state-of-the-art ME and ICE methods as our baselines for comparison with our STRUEDIT, which include the following approaches:

ROME ROME (Meng et al., 2022a) applies causal mediation analysis to locate the editing area, framing model editing as a least-squares problem under linear equality constraints and solving it using lagrange multipliers.

MEND MEND (Mitchell et al., 2021) adopt a meta-learning approach that trains a hypernetwork to infer weight updates from the gradient of the inserted fact.

MEMIT MEMIT (Meng et al., 2022b) insert new memories into language models by targeting key transformer weights identified as causal mediators of factual knowledge recall.

Model	Method	MQuAKE-3k	MQuAKE-2002	MQuAKE-hard
LLaMA2-7B-chat	ROME$^\spadesuit$ (Mitchell et al., 2021)	2.3	2.9	0.4
	MEMIT$^\spadesuit$ (Meng et al., 2022b)	3.1	3.5	0.6
	MEND$^\spadesuit$ (Meng et al., 2022a)	3.9	4.1	0.9
	IKE$^\diamondsuit$ (Zheng et al., 2023)	6.2	6.5	0.5
	MeLLo$^\diamondsuit$ (Zhong et al., 2023)	10.8	11.8	1.6
	DeepEdit$^\diamondsuit$ (Wang et al., 2024)	11.2	12.9	7.0
	StruEdit (ours)	**52.1**	**67.3**	**41.7**
LLaMA2-13B-chat	ROME$^\spadesuit$ (Mitchell et al., 2021)	3.1	4.8	0.7
	MEMIT$^\spadesuit$ (Meng et al., 2022b)	4.3	5.1	1.1
	MEND$^\spadesuit$ (Meng et al., 2022a)	4.8	5.3	1.3
	IKE$^\diamondsuit$ (Zheng et al., 2023)	6.8	7.7	1.2
	MeLLo$^\diamondsuit$ (Zhong et al., 2023)	11.2	12.3	1.5
	DeepEdit$^\diamondsuit$ (Wang et al., 2024)	12.5	13.7	8.2
	StruEdit (ours)	**53.4**	**68.5**	**48.9**

Table 2: Experimental results (accuracy; %) on MQuAKE datasets with open-source models. We conduct the experiments with the full batch size edit memory. Methods marked with $^\spadesuit$ belong to ME, while those marked with $^\diamondsuit$ belong to ICE. The best KE result on every LLM is highlighted in **bold** font.

IKE IKE (Zheng et al., 2023) uses demonstration contexts without parameter updates, prompting LLMs to perform edits by leveraging newly retrieved knowledge.

MeLLo MeLLo (Zhong et al., 2023) guides LLMs in multi-hop knowledge editing by decomposing subproblems and detecting conflicts between parametric knowledge and edited facts.

DeepEdit DeepEdit (Wang et al., 2024) enhances generating coherent reasoning chains with new knowledge through depth-first search.

4.3 Overall Performance

We set the edit memory to full batch size to reflect real-world scenarios in our experiments. Table 2 displays the KE performance of ME and ICE methods, as well as our StruEdit, on MQuAKE across open-source models. Overall, both ME and ICE methods perform poorly, while our StruEdit consistently shows a significant lead. ME methods rely on modifying model structures or parameters to update knowledge, which becomes inefficient when dealing with a large number of new knowledge instances, as in our full batch size experiment. This can negatively impact the model's inherent parametric knowledge and reasoning abilities. In the ICE methods, IKE struggles to retrieve relevant information from the vast amount of new knowledge, resulting in poor editing performance. Although MeLLo and DeepEditattempt to address this limitation through conflict detection and deep search, they are still constrained by the reasoning capabilities of open-source LLMs. Our StruEdit demonstrates significant improvement, highlighting its great potential in real-world scenarios.

The experimental results on the closed-source models are presented in Table 3. Due to the closed-source nature of the models, we are only able to test the ICE methods and our StruEdit. The ICE methods exhibit overall improvement compared to their KE performance on open-source models, as shown in Table 2, with DeepEditshowing the most significant gains. This improvement is attributed to the strong in-context learning (ICL) capabilities of the advanced closed-source models. However, our StruEdit also benefits from the enhanced ICL abilities of these LLMs, consistently achieving the best performance with an average accuracy that is 58.8% higher than DeepEdit.

Compared to more powerful closed-source models, StruEdit shows an even greater lead on smaller open-source models. This indicates that previous KE methods heavily relied on the inherent capabilities of LLMs, while StruEdit reduces the burden on LLMs during the KE process.

4.4 Robustness of Knowledge Editing

The robustness of KE is crucial for assessing whether knowledge has been thoroughly and effectively edited. We evaluate the robustness of knowledge editing by assessing it across both the number of hops in multi-hop QA and the number

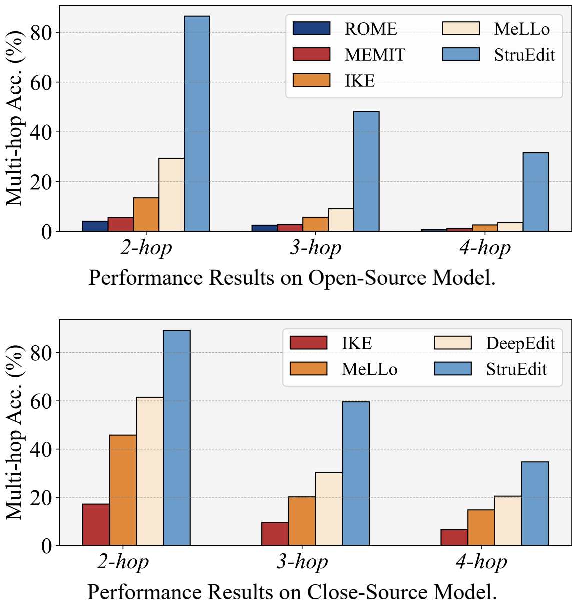
Model	Method	MQuAKE-3K	MQuAKE-2002	MQuAKE-hard
GPT-3.5 Turbo-Instruct	IKE$^\diamondsuit$ (Zheng et al., 2023)	10.2	12.5	1.4
	MeLLo$^\diamondsuit$ (Zhong et al., 2023)	20.0	25.1	1.6
	DeepEdit$^\diamondsuit$ (Wang et al., 2024)	38.0	48.0	53.7
	StruEdit (ours)	**62.7**	**85.3**	**75.5**
GPT-4o-mini	IKE$^\diamondsuit$ (Zheng et al., 2023)	10.5	13.1	1.5
	MeLLo$^\diamondsuit$ (Zhong et al., 2023)	21.2	27.8	2.4
	DeepEdit$^\diamondsuit$ (Wang et al., 2024)	41.3	49.2	55.8
	StruEdit (ours)	**66.5**	**86.3**	**77.3**

Table 3: Experimental results (accuracy; %) of ICE methods and our StruEdit on MQuAKE datasets with closed-source models. We conduct the experiments with the full batch size edit memory.

of edited instances.

Number of Hops in Multi-Hop QA Figure 5 shows the changes in KE performance for ME, ICE methods, and our StruEdit as the number of hops in multi-hop QA increases. We observed that, regardless of whether on open-source or closed-source models, all methods experience a noticeable decline in editing performance as the number of hops increases. This decline is primarily due to the hallucinations introduced by multi-hop reasoning and knowledge conflicts. Specifically, ME shows an average decrease of 57% and ICE an average decrease of 56% with each additional hop, whereas StruEdit only declines by 38% on average, demonstrating the strong robustness of our StruEdit with increasing reasoning hops.

Number of Edited Instances Edited instances refer to the number of new knowledge updates required in the edit memory. In real-world deployments, where the number of edited instances is often high, the robustness of KE in this aspect becomes especially critical. We conduct experiments based on randomly grouped edited instances of varying quantities, with the results shown in Figure 6. Consistent with the observations of Zhong et al. (2023) , all methods show further decline when more edits are injected. As the number of edited instances increases, both ME and ICE methods experience a significant decline, particularly in the comparison between 1-instance and 100-instance settings. This decline is primarily due to the challenges of imprecise knowledge localization, which fails to provide effective editing information, and hallucinations caused by knowledge conflicts. In contrast, our StruEdit consistently demonstrates the best performance across different instance scenarios, with the smallest average decline.



Performance Results on Open-Source Model.

Performance Results on Close-Source Model.

Figure 5: Multi-hop QA results across 2, 3, and 4 hops on both open-source (LLaMA2-7B-Chat) and closed-source (GPT-3.5-Turbo-Instruct) models for ME, ICE methods, and our StruEdit.

Overall, the performance of existing KE methods is significantly impacted by more complex multi-hop reasoning and a higher number of edited instances, which implies that many pieces of knowledge may not be thoroughly edited in real-world scenarios. In contrast, our StruEdit demonstrates more stable robustness by reasoning over the updated knowledge structure, effectively mitigating hallucinations caused by retrieval errors, reasoning challenges, and knowledge conflicts.

4.5 Editing Latency

Table 4 shows the latency of different editing methods in multi-hop QA. ME methods are generally slower, as it requires locating and modifying the model based on the edited facts before reasoning. ICE methods are faster, but latency increases when

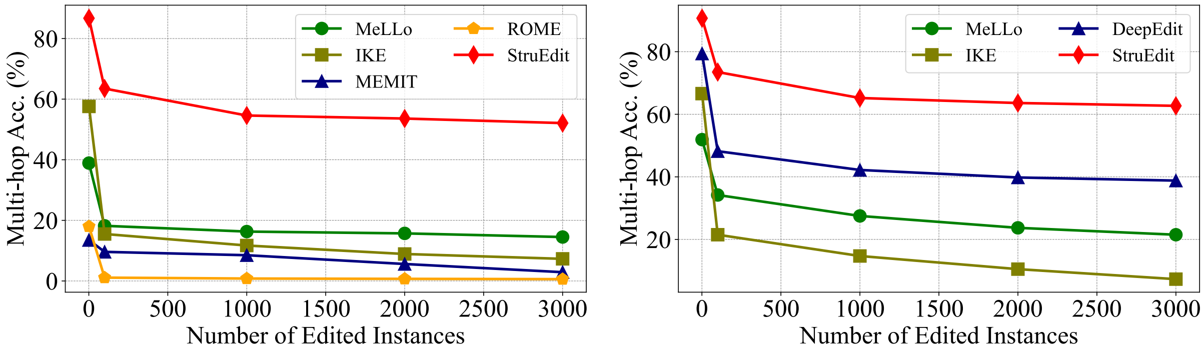


Figure 6: Multi-hop performance (CoT) of LLAMA2-7B-CHAT (left) and GPT-3.5-TURBO-INSTRUCT (right) across different KE methods with 1, 100, 1000, 2000, and 3000 edited instances drawn for editing

longer text reasoning is needed to improve editing performance, as seen in methods like MeLLo and DEEPEDIT. STRUEDIT demonstrates the best efficiency, even compared to the simplest IKE method, because it achieves strong reasoning performance without relying on LLMs generating lengthy CoT, thanks to the support of knowledge structures.

Method	*2-hop*	*3-hop*	*4-hop*
MEND♠	91.7	126.3	167.2
ROME♠	43.7	52.3	63.8
MEMIT♠	113.6	155.2	213.4
IKE◇	2.48	2.67	2.85
MeLLo◇	12.83	16.27	21.53
DEEPEDIT◇	10.05	15.79	19.58
STRUEDIT	**1.75**	**2.25**	**2.38**

Table 4: Average latency (s/QA) for KE on MQUAKE-2002 with LLAMA2-7B-CHAT. The lowest latency is highlighted in **bold**. ME latency includes model editing and reasoning, while ICE latency includes knowledge retrieval and reasoning.

5 Related Work

LLMs' Hallucination Pre-training on large-scale corpora equips LLMs with extensive parametric memory, including commonsense and factual knowledge (Petroni et al., 2019; Li et al., 2022). However, this parametric knowledge may be inaccurate due to errors or outdated information in the pre-training data, leading to hallucinations (Zhang et al., 2023; Huang et al., 2023a; Wang et al., 2023a) where the content generated by LLMs deviates from established world knowledge.

Knowledge Conflict To mitigate the hallucinations, tools (Nakano et al., 2022; Yao et al., 2023a; Qin et al., 2024) or retrieval-augmented methods (Guu et al., 2020; Izacard and Grave, 2021; Zhong et al., 2022), such as ChatGPT Plugins and New Bing, have been proposed as effective solutions to provide external knowledge evidence. However, external knowledge may inevitably conflict (Petroni et al., 2020; Si et al., 2023; Xie et al., 2024) with parametric knowledge, leading to unreliable support, especially when LLMs are overly confident in their own parametric knowledge.

Knowledge Editing KE (Yao et al., 2023b) has been proposed to update outdated information, enabling models to answer current questions accurately. In general, existed KE can be divided into two main categories. ME (Zhu et al., 2020; Meng et al., 2022a,b; Huang et al., 2023b) involves modifying model parameters or structure to prevent undesired outputs. ICE (Mitchell et al., 2022; Madaan et al., 2022; Zhong et al., 2023; Zheng et al., 2023) edit knowledge by prompting LLMs with the newly updated facts. However, both approaches are affected by localization or knowledge conflicts, leading to hallucinations.

6 Conclusion

In this paper, we proposed a new improved baseline for knowledge editing, called STRUEDIT. Unlike the locate-and-edit KE approaches such as ME and ICE, STRUEDIT removes all parametric knowledge, regardless of whether it conflicts with new knowledge. By leveraging LLMs to extract entities and relations from the original question, STRUEDIT performs multi-hop reasoning over up-to-date knowledge structures to derive accurate answers. This new paradigm offers higher editing accuracy, faster performance, and greater robustness. Our work paves the way for further advancements in knowledge editing.

Limitations

This work presents an improved baseline for KE. Unlike ME and ICE, which rely on text-based editing information, STRUEDIT requires a more structured knowledge format to support LLM reasoning. The decline from the editing tasks in Table 1 to the QA tasks in Tables 2 and 3 reflects the loss in entity and relation extraction for STRUEDIT. While STRUEDIT demonstrates strong robustness, there is still a noticeable decline as reasoning hops and the number of edited instances increase, indicating potential errors in LLM reasoning.

Ethical Considerations

In this study, we adhere to ethical guidelines by using only open-source datasets and employing models that are either open-source or well-established in the scientific community. We utilize counterfactual public datasets for knowledge editing to evaluate knowledge updates. Our proposed STRUEDIT method focuses on updating knowledge to enable LLMs to accurately answer real-world questions. We are committed to maintaining high ethical standards throughout our research, emphasizing transparency and promoting the responsible use of technology for the betterment of society.

References

Baolong Bi, Shenghua Liu, Lingrui Mei, Yiwei Wang, Pengliang Ji, and Xueqi Cheng. 2024a. Decoding by contrasting knowledge: Enhancing llms' confidence on edited facts. *Preprint*, arXiv:2405.11613.

Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, and Xueqi Cheng. 2024b. Is factuality decoding a free lunch for llms? evaluation on knowledge editing benchmark. *arXiv preprint arXiv:2404.00216*.

Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, and Xueqi Cheng. 2024c. LPNL: Scalable link prediction with large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3615–3625, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, Hongcheng Gao, Yilong Xu, and Xueqi Cheng. 2024d. Adaptive token biaser: Knowledge editing via biasing key entities. *arXiv preprint arXiv:2406.12468*.

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

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *Preprint*, arXiv:2002.08909.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *Preprint*, arXiv:2311.05232.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023b. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. *Preprint*, arXiv:2007.01282.

Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2022. A systematic investigation of commonsense knowledge in large language models. *Preprint*, arXiv:2111.00607.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve gpt-3 after deployment. *arXiv preprint arXiv:2201.06009*.

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

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*.

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

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin

Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *Preprint*, arXiv:2112.09332.

OpenAI. 2022. large-scale generative pre-training model for conversation. *OpenAI blog*.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. *Preprint*, arXiv:2005.04611.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *Preprint*, arXiv:1909.01066.

Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, Yi Ren Fung, Yusheng Su, Huadong Wang, Cheng Qian, Runchu Tian, Kunlun Zhu, Shihao Liang, Xingyu Shen, Bokai Xu, Zhen Zhang, Yining Ye, Bowen Li, Ziwei Tang, Jing Yi, Yuzhang Zhu, Zhenning Dai, Lan Yan, Xin Cong, Yaxi Lu, Weilin Zhao, Yuxiang Huang, Junxi Yan, Xu Han, Xian Sun, Dahai Li, Jason Phang, Cheng Yang, Tongshuang Wu, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2024. Tool learning with foundation models. *Preprint*, arXiv:2304.08354.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2023. Prompting gpt-3 to be reliable. *Preprint*, arXiv:2210.09150.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks. *arXiv preprint arXiv:2004.00345*.

Zezheng Song, Jiaxin Yuan, and Haizhao Yang. 2024. Fmint: Bridging human designed and data pretrained models for differential equation foundation model. *arXiv preprint arXiv:2404.14688*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023a. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, et al. 2023b. 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. 2024. Deepedit: Knowledge editing as decoding with constraints. *arXiv preprint arXiv:2401.10471*.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. *Preprint*, arXiv:2305.13300.

Derong Xu, Ziheng Zhang, Zhihong Zhu, Zhenxi Lin, Qidong Liu, Xian Wu, Tong Xu, Xiangyu Zhao, Yefeng Zheng, and Enhong Chen. 2024. Editing factual knowledge and explanatory ability of medical large language models. *arXiv preprint arXiv:2402.18099*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023a. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023b. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *Preprint*, arXiv:2309.01219.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

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? *arXiv preprint arXiv:2305.12740*.

Zexuan Zhong, Tao Lei, and Danqi Chen. 2022. Training language models with memory augmentation. *Preprint*, arXiv:2205.12674.

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*.