CompKE: Complex Question Answering under Knowledge Editing

Anonymous ACL submission

Abstract

Knowledge Editing—Efficiently modifying the knowledge in large language models has gathered great attention. Current benchmarks primarily use multi-hop question answering to assess and analyze newly injected or updated knowledge. However, we argue that these benchmarks fail to effectively evaluate how well the updated models apply this knowledge in real-life scenarios, particularly when questions require complex reasoning, involving one-to-many relationships or multi-step logical intersections. To fill in this gap, we introduce a new benchmark, CompKE: **Comp**lex Question Answering under **K**nowledge **E**diting, which includes 11,924 complex questions that reflect real-life situations. We perform a comprehensive evaluation of four different knowledge editing methods on CompKE, and our results show that the performance of these methods varies between different models. For example, MeLLo achieves an accuracy of 39.47 on GPT-4o-mini but drops significantly to 3.83 on Qwen2.5-3B. We further analyze the reasons behind these results from both methodological and model perspectives. Our dataset will be publicly available on GitHub.

1 Introduction

Despite large language models (LLMs) are powerful in solving a wide range of real-world scenarios, they often generate outdated or incorrect knowledge (Wang et al., 2023b; Zhang et al., 2024b). Therefore, Knowledge Editing (KE), *i.e.,* updating the model's knowledge by avoiding expensive fine-tuning, has become an active research domain (Wang et al., 2023b; Zhang et al., 2024b).

To comprehensively evaluate the effectiveness of KE methods, a common approach is to assess whether the model can reproduce the newly injected knowledge, as demonstrated in ZsRE (Levy et al., 2017) and COUNTERFACT (Meng et al., 2022a). However, these benchmarks cannot determine whether the model genuinely utilizes the newly injected knowledge or simply memorizes and regurgitates it. MQuAKE (Zhong et al., 2023) addresses this issue through multi-hop question answering (MQA). An example in this regard is illustrated in Figure 1 (a), which shows a question: *"Who is the spouse of the president of U.S.?"* This question requires multiple reasoning steps: *(a)* identifying who is the current president of U.S.; and, *(2)* determining the president's spouse.

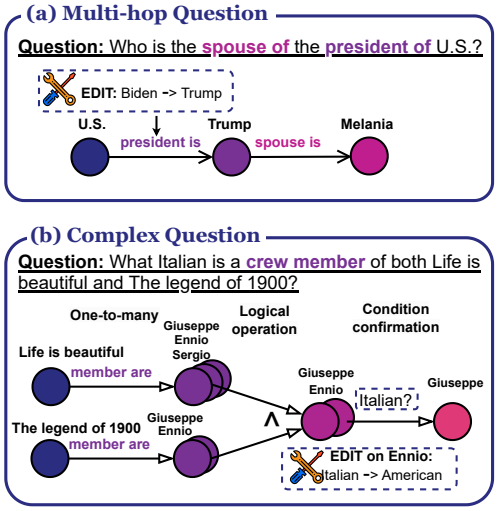


Figure 1: (a) An example of a multi-hop question involving only one-to-one sequential step-by-step reasoning. (b) An example of a complex problem involving one-to-many knowledge mapping, logical operations, and conditional confirmation.

However, the evaluation dimensions of multi-hop questions remain too narrow to fully assess the model's ability to flexibly apply the newly integrated knowledge. This limitation manifests itself in three key areas: *(i) linear question structure*, the questions follow a strict pattern, resulting in an overly simplistic structure that can be arranged in a linear sequence. *(ii) one-to-one relations*, each fact triple in the sub-questions adheres strictly to a one-to-one relation, which fails to reflect real-world knowledge representations. In prac-

tice, many facts involve one-to-many relationships, such as *"Who are the major shareholders of a company?"*—where a single subject is linked to multiple entities. *(iii) limited edit operations*, the knowledge edits are limited to substitutions, overlooking more complex real-world modifications.

To address this gap, we propose a new benchmark for complex questions, *i.e.,* CompKE: **Comp**lex Question Answering under **K**nowledge **E**diting. CompKE, originally derived from Wikidata, comprises 11,924 complex questions. As shown in Figure 1 (b), compared to the multi-hop knowledge editing benchmark, CompKE offers the following advantages:

(i) Diverse structures: Individual sub-questions in CompKE are combined in multiple ways to create complex questions, integrating logical operations, conditional verification, and knowledge mapping.

(ii) One-to-many relations: The fact triples that formulate complex problems encompass both one-to-one and one-to-many relations.

(iii) Expanded capabilities: CompKE systematically incorporates real-world knowledge updates, extending beyond simple substitutions to encompass additions and deletions.

We perform a comprehensive evaluation of major KE methods across five LLMs from different model families, including both open-source and closed-source architectures with varying parameter sizes. The results show that most methods demonstrate relatively low performance. Additionally, we analyze the effectiveness of each method across models with different parameter scales. Our findings suggest that parameter-based approaches are more effective for smaller models, whereas memory-based methods achieve better results in larger models with stronger reasoning capabilities. We summarize the key contributions of our work as follows:

- We introduce CompKE, a novel KE benchmark that overcomes existing limitations by incorporating diverse question structures, one-to-many relations, and expanded edit types.
- We comprehensively evaluate major KE methods across five LLMs, uncovering significant differences in their ability to handle complex logical problems in diverse KE scenarios and providing an in-depth analysis of the underlying factors.

2 Related Work

Knowledge Editing Benchmarks. KE is a crucial research area for LLMs, enabling them to update information and adapt to evolving real-world queries. Various benchmarks have been established to assess the effectiveness of KE methods.

Early works like COUNTERFACT (Meng et al., 2022a) assess counterfactual updates, while ZsRE (Levy et al., 2017) and MzsRE (Wang et al., 2023c) extend evaluations to zero-shot and multilingual settings. ECBD (Onoe et al., 2023) examines whether newly injected facts can propagate reasoning across related entities. Easyedit (Wang et al., 2023a) propose an easy-to-use framework for LLMs that supports a variety of cutting-edge KE approaches. More recent works such as MQuAKE (Zhong et al., 2023), MQA-AEVAL (Ali et al., 2024) extend the evaluation to multi-hop reasoning under KE. TEMPLAMA (Zheng et al., 2023a) and ATOKE (Yin et al., 2023) explore the task of time-series knowledge editing, aiming to modify knowledge without affecting knowledge from other time periods. However, these benchmarks fall short in capturing real-world complexity, such as reasoning with one-to-many relations or combining entities via logical operations such as intersection and union.

Knowledge Graph Question Answering. There exist several complex question-answering datasets in the Knowledge Graph (KG) domain. ComplexQuestions (Bao et al., 2016) evaluates KG-based systems' ability to handle multi-constraint queries. MetaQA (Zhang et al., 2018) is a multi-hop dataset in the movie domain, incorporating both textual and audio data and requiring reasoning over up to three hops. ComplexWebQuestions (Talmor and Berant, 2018), built on the Freebase knowledge base, involves answering complex questions by reasoning across multiple web snippets. CR-LT-KGQA (Guo et al., 2024) focuses on commonsense reasoning and long-tail knowledge. Although complex questions have been extensively studied in the KG domain, they cannot be directly applied to the knowledge editing field due to two key challenges: *(i) Omission of sub-questions* and *(ii) Knowledge dependency*. A detailed explanation is provided in Appendix A.2.

Knowledge Editing Methods. We sub-divide existing research on KE into parameter-based and memory-based methods.

Parameter-based KE methods aim to directly modify the model's internal parameters to reflect updated knowledge. For example, ROME (Meng et al., 2022a) and MEMIT (Meng et al., 2022b) focus on identifying and modifying parameters associated with specific knowledge, while Transformer-Patcher (Huang et al., 2023) edits facts by adding neurons. To reduce computational costs and prevent catastrophic forgetting, techniques such as: LoRA (Hu et al., 2021), Prompt Tuning (Shi and Lipani, 2024), and QLoRA (Dettmers et al., 2023) have been proposed.

However, after KE, these methods struggle with multi-hop and complex questions and cannot be applied to closed-source models like OpenAI GPTs, which are accessible only via APIs. Moreover, they are more computationally expensive than memory-based approaches.

Memory-Based methods store updates in external memory and retrieve them as needed during inference. For instance, SERAC (Mitchell et al., 2022) combines semi-parametric editing with retrieval augmented counterfactual models for efficient knowledge updates. GRACE (Hartvigsen et al., 2022) integrates adapters into LLMs and uses vector matching to modify knowledge entries. IKE (Zheng et al., 2023b) applies in-context learning with stored demonstrations for knowledge modification, MeLLo (Zhong et al., 2023) stores edited facts externally and utilizes prompts to incorporate edits during inference. PokeMQA (Gu et al., 2023) separates question decomposition and conflict detection using a two-stage programmable scope detector. GLAME (Zhang et al., 2024a) employs a knowledge graph module to enhance retrieval efficiency.

We observed, MeLLo and PokeMQA excel at multi-hop problems, therefore in our experiments, we use them as baselines to assess the generalization of memory-based methods to complex questions. We provide further details about related work in Appendix A.

3 Preliminaries

We use $\mathcal{D} = \{(s, r, o)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ to denote the set of knowledge triplets, where $\mathcal{E}$ and $\mathcal{R}$ denote the set of entities and relations respectively. Each triple (s, r, o) represents a knowledge instance, implying that the subject entity s and the object entity o are related by relation r. In order to represent one-to-many knowledge instances, we expand the original definition of knowledge instance to $(s, r, \mathcal{O})$, where $\mathcal{O} = \{o_1, o_2, \cdots\}$ is a set of object entities, *e.g.,* (Avatar, actors_are, {Worthington, Saldana,$\cdots$}).

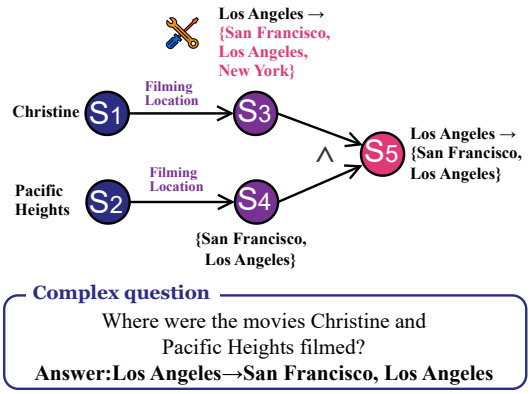


Figure 2: An example of a complex question under knowledge editing. Knowledge editing occurs in the first sub-question, where the filming location of *Christime* is modified from { Los Angeles} to {San Francisco, Los Angeles, New York}.

3.1 Complex Questions

Building on the example in the introduction, we formally define the complex questions to be explored in this paper. A brief recap of multi-hop question answering (MQA) and MQA under KE is provided in the Appendix B.1. We define a complex question Q as a question that could be represented as a graph-like reasoning structure, *i.e.,* $Q = (\mathbf{S}, \mathbf{L})$, where $\mathbf{S} = \{S_1, S_2. \cdots\}$ represents a set of *intermediate entities* and $\mathbf{L} = \{L_1, L_2, \cdots\}$ denotes a set of *reasoning links*. Each $S_i \in \mathbf{S}$ is a set of entities, *i.e.,* $S_i = \{s_1, \cdots\}$, used to represent one-to-one and one-to-many knowledge instances. Each $L_i \in \mathbf{L}$ is a reasoning link. Note that unlike the relation typically used in knowledge graphs (used to map one entity s_i to another s_j via the relation r), reasoning links offer extended operations by allowing conditional confirmation and logical operations, which are formally explained below.

Reasoning Links. We categorize the reasoning links into two distinct categories:

***(a)* Knowledge-related Links:** These links facilitate entity traversal along the link, *e.g.,* given a set of entities $S_i \in \mathbf{S}$, a reasoning link may be used to obtain the next step entities $S_j \in \mathbf{S}$. We subdivide these links into:

(i) Knowledge Mapping. For S_i, we consider

a knowledge mapping link as a mapping to the set of adjacent entities $S_j = \cup_{s \in S_i} A_r(s)$, where $A_r(s) = \{s' \mid (s, r, s') \in \mathcal{D}\}$ represents the entities related to s via relation r.
(ii) Condition Confirmation. Given r and s', this link aims to identify a set of entities $S_j = \{s \in S_i \mid C(s, r, s') = \texttt{True}\}$ conforming to the condition $[C(s, r, s') \in \mathcal{D}]$, which is used to examine whether s can obtain s' via r.

***(b)* Logical Links:** Given a set of intermediate entities $\{S_1, S_2, \cdots, S_n\} \in \mathbf{S}$, these reasoning links perform logical operations among the elements of S_i. Specifically, we use logical links for the following operations:
(i) Intersection. The intersection operation identifies adjacent entities shared across all sets, *i.e.,* $S_j = \cap_{k=1}^{n} S_k$.
(ii) Union. The union operation computes the set of adjacent entities, including all entities from any of the sets, *i.e.,* $S_j = \cup_{k=1}^{n} S_k$.

Example 1. An example of a complex question with reasoning links is illustrated in Figure 2. It shows the question: *"Where were the movies Christine and Pacific Heights filmed?"*. The intermediate entities are S_1={Christine}; S_2={Pacific Heights}; S_3= {Los Angeles}; S_4={San Franciso, Los Angeles}; and S_5={Los Angeles}. The reasoning operations are: $L_1 : S_1 \xrightarrow{\text{filming\_at}} S_3$; $L_2 : S_2 \xrightarrow{\text{filming\_at}} S_4$; followed by L_3 : logical operation on S_3 and S_4 ($S_3 \cap S_4$) to obtain the final answer, *i.e.,* S_5 = {Los Angeles}.

Complex Question Answering under KE. We use $e = (s, r, \mathcal{O} \rightarrow \mathcal{O}')$ to represent knowledge editing for one-to-many instances showing that $\mathcal{O}$ is updated to $\mathcal{O}'$. The task assumes that the language model has access to original knowledge base $\mathcal{D}$. Given a batch of edits $\mathcal{E} = \{e_1, e_2, \cdots\}$, the knowledge to be deleted as $\mathcal{D}^{\mathcal{E}}_{del} = \{(s_i, r_i, \mathcal{O}_i) \mid e_i \in \mathcal{E}\}$, and the newly added knowledge as $\mathcal{D}^{\mathcal{E}}_{add} = \{(s_i, r_i, \mathcal{O}'_i) \mid e_i \in \mathcal{E}\}$, the end-goal is to update the LLM's knowledge by $\mathcal{D}'$, define as: $\mathcal{D}' = (\mathcal{D} - \mathcal{D}^{\mathcal{E}}_{del}) \cup \mathcal{D}^{\mathcal{E}}_{add}$. Finally, updated knowledge $\mathcal{D}'$ is used to generate the final answer to the complex question Q.

4 CompKE

While complex questions are common in real-life scenarios, they remain underexplored in LLM question answering under KE. We argue that existing benchmarks predominantly focus on linear multi-hop questions, limiting their effectiveness in evaluating complex queries. To bridge this gap, we propose CompKE: **Comp**lex Question Answering under **K**nowledge **E**diting. In the following, we provide a brief overview of CompKE followed by a detailed flow of the process.

4.1 Dataset Construction

Overview. The workflow of our data construction process, illustrated in Figure 3, follows six key steps. First, we extract factual triples from Wikidata. Next, we select relevant relations and sample triples corresponding to those relations. In the third step, these triples are combined into complex questions with diverse reasoning structures, and edits are introduced at appropriate positions within the questions. This is followed by the introducing counterfactual modifications in step four and the filtering of conflicting instances in step five to maintain consistency. Finally, in step six, the structured questions are converted into natural language. Further details on each step are provided below.

Step 1: Collecting Relation Templates. In step 1, we carefully select 37 relations from Wikidata's *List of Properties*, through a two-step process. First, we identify essential one-to-many relations (e.g., family-child, book-authors, movie-actors) for one-to-many knowledge mapping. Next, we incorporate one-to-one relations (e.g., country-capital, person-hometown) that capture fundamental entity attributes, enabling one-to-one knowledge mapping and conditional confirmation. In addition, we prioritize relations commonly encountered in everyday scenarios to enhance the practical utility of the data set for the relevance of the real world. The full list of relation templates used in CompKE is provided in Appendix Table 9.

Step 2: Sampling Facts. After selecting relation templates, we build the knowledge base $\mathcal{D}$ with a focus on commonly known rather than obscure knowledge. Using the collected relation templates, we sample single-hop knowledge triples from Wikidata and rank them by access frequency, prioritizing the most frequently accessed triples. To refine this selection, we employ GPT-3.5-Turbo-Instruct to filter out knowledge the model cannot recall. The finalized knowledge base $\mathcal{D}$ serves as the basis for generating complex questions.

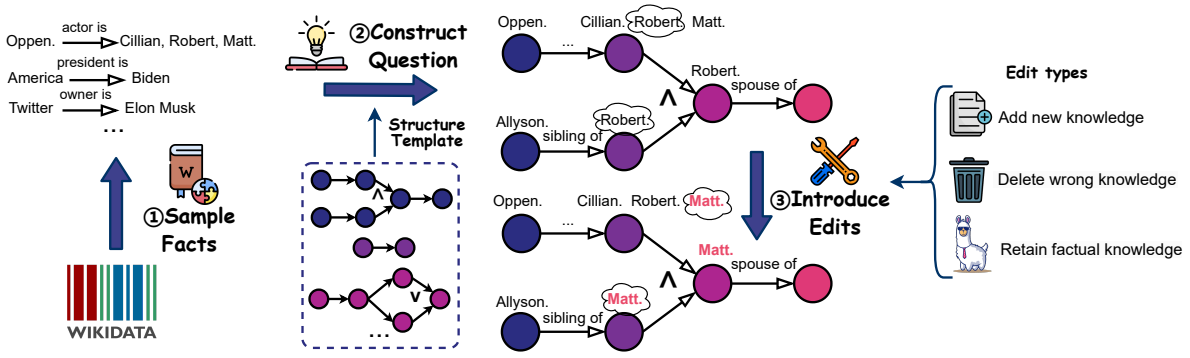


Figure 3: The construction process of CompKE.

Step 3: Constructing Complex Questions. We observe that complex questions often follow structured reasoning patterns, as shown in Figure 2, where knowledge mapping is followed by logical operations (*e.g.,* intersection). To systematically collect these reasoning structures, we first manually curate a high-quality subset of complex questions as seed examples. We then extract their underlying reasoning structures by removing intermediate entities, forming reusable templates. These templates are instantiated with real-world facts from $\mathcal{D}$ to generate specific complex questions. The process begins with the random initialization of the leaf nodes, followed by the iterative identification of intermediate entities using logical operations or knowledge of $\mathcal{D}$, continuing until all entities are fully determined.

To ensure the practical relevance of instantiated questions, we filter out cases that meet the following criteria: *(i)* questions with no valid answer, *(ii)* questions yielding an empty set of intermediate entities, and *(iii)* cases where entities involved in logical operations are of incompatible types. For illustration, exemplar relational structures and their corresponding instantiated complex questions are provided in Appendix (Figure 10).

Step 4: Introducing Counterfactual Edits. To simulate knowledge edits, we construct counterfactual knowledge updates. For each complex question, we randomly select knowledge mappings with knowledge of the form: $(s, r, \mathcal{O})$ and introduce an edit $e = (s, r, \mathcal{O}')$. Unlike previous benchmarks that only involve one-to-one relations with edits limited to entity replacement, our dataset introduces edits that involve one-to-many relations, leading to more complex edits. To clearly represent the changes in a fact triple, we define three basic operations, each of which can be combined to form an edit:

$\mathcal{E}$	(*Christine*, filming location, { Los Angeles}→ {San Francisco, Los Angeles, New York})
$\mathcal{Q}$	*i)Where were the movies Christine and Pacific Heights filmed?* *ii)*In which locations were both the movie *Christine* and *Pacific Heights* filmed? *iii)*What were the filming locations for both the movie *Christine* and *Pacific Heights*?
$\mathcal{A}$	{Los Angeles}
$\mathcal{A}^*$	{San Francisco, Los Angeles}
$\mathcal{T}$	(*Christine*, filming location,{Los Angeles}) (*Pacific Heights*, filming location,{San Francisco, Los Angeles})
$\mathcal{T}^*$	(*Christine*, filming location,{San Francisco, Los Angeles, New York}) (*Pacific Heights*, filming location,{San Francisco, Los Angeles})
$\mathcal{L}$	{Los Angeles} ∩ {San Francisco, Los Angeles} = {Los Angeles}
$\mathcal{L}^*$	{San Francisco, Los Angeles, New York} ∩ {San Francisco, Los Angeles} ={San Francisco, Los Angeles}

Table 1: A case from CompKE, illustrating the components involved in question editing. Here, $\mathcal{E}$ represents the edit, $\mathcal{Q}$ is the natural language question, $\mathcal{A}$ and $\mathcal{A}^*$ denote the answers before and after editing respectively. $\mathcal{T}$ and $\mathcal{T}^*$ are the sets of fact triples before and after editing, which form the complex question. Additionally, $\mathcal{L}$ and $\mathcal{L}^*$ indicate the logic operations applied to the question before and after editing.

(i) Addition: $\mathcal{O}_{\text{add}} = \mathcal{O}' \setminus \mathcal{O}$, where $\mathcal{O}_{\text{add}}$ represents the set of newly added entities;
(ii) Deletion: $\mathcal{O}_{\text{del}} = \mathcal{O} \setminus \mathcal{O}'$, where $\mathcal{O}_{\text{del}}$ represents the set of removed entities;
(iii) Retention: $\mathcal{O}_{\text{ret}} = \mathcal{O} \cap \mathcal{O}'$, where $\mathcal{O}_{\text{ret}}$ represents the set of retained entities.

Example 2. An example of an edit to change the management of the "Microsoft" may be expressed as: (Microsoft, managers_are, {John, Smith, Dave} → {Smith, Eden, Keyes}), which involves deleting {John}, retaining {Smith}, and adding {Eden, Keyes}.

Step 5: Filtering Conflicting Edits. Since the counterfactual edits in Step 4 are introduced randomly, for a batch of edits $\mathcal{E} = \{e_1, e_2, \dots\}$ there may be edits corresponding to different cases where $e_i = (s_i, r_i, \mathcal{O}_i \rightarrow \mathcal{O}_i^*)$ and $e_j = (s_j, r_j, \mathcal{O}_j \rightarrow \mathcal{O}_j^*)$, with $s_i = s_j$ and $r_i = r_j$, but $\mathcal{O}_i^* \neq \mathcal{O}_j^*$. This indicates the presence of

conflicting facts within the batch. Simultaneously introducing such conflicts can severely compromise the validity evaluation. To mitigate this, we identify and group conflicting cases, and then randomly select only one to retain.

Step 6: Phrasing in Natural Language. Building on steps 1–5, we construct complex questions involving edits, each comprising multiple fact triples. To facilitate evaluation by the target LLMs, these questions must be translated into natural language. For each reasoning structure defined in Step 3, we first manually curate eight high-quality examples. Then, using GPT-4o-mini, we generate three natural language questions for each structured question. Further details on constructing the dataset are provided in the Appendix C.

4.2 Dataset Summary

Table 2 presents the dataset distribution across two dimensions: `Edit_num` and `Step_num`. `Edit_num` represents the number of triples edited in a complex question. In CompKE, most cases involve editing a single triple, followed by cases with two edits. `Step_num` denotes the number of reasoning steps required to answer the complex question, with 3-step questions being the most prevalent, followed by 4-step and 5-step questions respectively.

Example 3. Table 1 presents a detailed example from CompKE, illustrating a complex question constructed by combining two sub-questions with an intersection operation. We assume that the editing takes place in the first sub-question (i.e., Christine's filming locations are updated from {Los Angeles} to {San Francisco, Los Angeles, New York}), leading to the addition of San Francisco in the final answer.

5 Experiments

In this section, we conduct a comprehensive evaluation of recent knowledge editing methods in CompKE, assessing them from three aspects: whether newly added knowledge can be recalled, whether existing knowledge is retained, and overall accuracy. We also analyze how the performance of different methods changes when the edit batch size (*i.e.,* the number of edits performed at once) increases. Additionally, by case studies, we observe several interesting phenomena, including overfitting in parameter-based methods, model collapse when increasing edit batch size, and the omission phenomenon in memory-based methods.

#Edits	1	2	3	4	5	Total
`Edit_num`	9,697	1,118	998	103	8	11,924
`Step_num`	200	424	5,770	2,949	2,581	11,924

Table 2: Statistical Results of CompKE.

5.1 Experimental Settings

Language Models. We conduct experiments using five different target LLMs corresponding to three model families. For open source models, we select LLaMA-3.1-8B-Instruct (Abhimanyu Dubey et al., 2024), Qwen2.5-3B-Instruct (Team, 2024), Qwen2.5-7B-Instruct (Team, 2024). For closed source models, we select GPT-3.5-turbo and GPT-4o-mini (Achiam et al., 2023).

Baselines. For performance comparison, we use the best performing methods for MQA under KE as baselines. These include parameter-based variants: ROME (Meng et al., 2022a), and MEMIT (Meng et al., 2022b); and memory-based variants: MeLLo (Zhong et al., 2023), and PokeMQA (Gu et al., 2023). Since GPT-3.5-turbo and GPT-4o-mini can only be accessed through APIs, parameter-based knowledge editing methods cannot be applied to them.

Evaluation Metrics. We use the following metrics for evaluation:

(i) Augment Accuracy (`Aug`): The number of newly introduced entities added to the answer list after the knowledge edit that are correctly identified compared to the original list.

(ii) Retain Accuracy (`Ret`): The number of entities present in both the original and edited answer lists, indicating the model's ability to preserve unmodified knowledge.

(iii) Accuracy (`Acc`): The average of `Aug` and `Ret`, offering a holistic measure of the model's accuracy in answering complex questions under KE.

The detailed mathematical formulations of these metrics are provided in Appendix D.3.

Example 4. Following the example in Figure 2, the final answer changes from {Los Angeles} before editing to {San Francisco, Los Angeles} after editing. `Aug` evaluates whether the model correctly includes the newly added entity, {San Francisco}, while `Ret` assesses its ability to retain

Model	Method	1-edited			100-edited			3000-edited		
		Aug	Ret	Acc	Aug	Ret	Acc	Aug	Ret	Acc
QWEN2.5-3B	ROME	12.61	17.91	15.26	4.80	4.40	4.60	0.82	1.59	1.21
	MEMIT	**20.99**	**23.86**	**22.43**	**7.80**	**6.73**	**7.27**	**1.52**	**3.75**	**2.64**
	MeLLo	5.40	2.25	3.83	3.06	3.39	3.23	0.69	2.00	1.35
	PoKeMQA	4.26	1.85	3.06	2.85	1.38	2.12	0.71	0.62	0.67
QWEN2.5-7B	ROME	22.82	25.09	23.96	7.50	7.98	7.74	0.73	0.98	0.86
	MEMIT	**29.40**	**27.72**	**28.56**	**24.11**	**24.80**	**24.46**	1.88	2.05	1.97
	MeLLo	17.78	13.38	15.58	10.35	17.32	13.84	**8.98**	**12.59**	**10.79**
	PoKeMQA	15.59	11.41	13.50	8.17	13.67	10.92	5.04	9.15	7.10
LLAMA-3.1-8B	ROME	7.44	24.84	16.14	1.50	1.14	1.32	0.56	0.61	0.59
	MEMIT	4.90	**33.22**	**19.06**	5.00	**29.27**	**17.14**	5.03	**29.20**	**17.12**
	MeLLo	**14.06**	17.95	16.00	**9.17**	17.84	13.51	**8.98**	14.17	11.58
	PoKeMQA	11.40	15.10	13.25	8.87	16.85	12.86	7.45	12.73	10.09
GPT-3.5-TURBO	MeLLo	**49.21**	**44.88**	**47.05**	**37.10**	**44.09**	**40.60**	**32.61**	**38.58**	**35.60**
	PoKeMQA	23.20	25.15	24.18	21.47	23.28	22.38	20.20	22.20	21.20
GPT-4O-MINI	MeLLo	22.07	25.19	23.63	20.31	23.62	21.96	18.75	22.14	20.45
	PoKeMQA	**36.60**	**42.33**	**39.47**	**35.42**	**41.35**	**38.39**	**28.36**	**35.02**	**31.69**

Table 3: Experimental results for COMPKE. We **boldface** the best results.

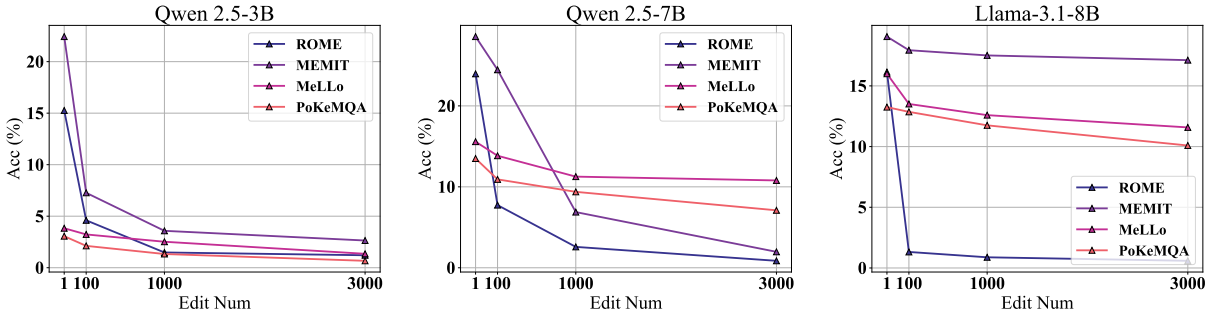


Figure 4: Variation of *Accuracy* (Acc) across QWEN2.5-3B, QWEN2.5-7B, and LLAMA-3.1-8B models with varying edit numbers. Results for GPT-3.5-TURBO and GPT-4O-MINI are provided in Appendix E.2.

the existing entity, {Los Angeles}, which appears both before and after editing. Acc, as the average of Aug and Ret, measures the model's effectiveness in integrating new knowledge while preserving existing information.

Experiment Setup. We conduct experiments on varying scales of knowledge edits, *i.e.,* using a batch of k-edits at a time with $k = \{1, 100, 1000, 3000\}$. To ensure a fair comparison with existing memory-based methods, we use the decomposition examples of complex questions for MeLLo and PokeMQA, as prompts. Additional details on the experimental setting are provided in Appendix D.

5.2 Experimental Results

The experimental results are summarized in Table 3. In general, MeLLo achieves the highest performance in the 1-edit setting on GPT-3.5-Turbo, yielding Aug score = 49.21. When comparing different approaches, memory-based methods perform poorly on smaller models (*e.g.,* QWEN2.5-3B) due to their reliance on the instructions' following and reasoning capabilities. In contrast, parameter-based methods are more effective for smaller models, but suffer substantial performance degradation as the edit batch size increases. In the following, we provide a detailed analysis of these findings.

Batch Editing (#k-edits). Figure 4 illustrates the accuracy of the four methods on QWEN2.5-3B, QWEN2.5-7B, and LLAMA-3.1-8B with an increase in the number of edits. Performance variations for GPT-3.5-TURBO and GPT-4O-MINI are provided in Appendix E.2.

We observe that memory-based methods experience a gradual decline in performance as the number of edits (k) increases. In contrast, parameter-based methods degrade more rapidly, especially when the number of edits exceed a certain thresh-

old. Notably, when $k \geq 100$, the model loses coherence, producing inconsistent responses and generating irrelevant outputs, as detailed in the Appendix Figure 9.

Smaller Models. For smaller models, such as QWEN2.5-3B, memory-based methods perform poorly compared to parameter-based methods. This can be attributed to two key factors: (i) Smaller models have limited instruction-following capabilities and struggle to adhere to the required format for response planning. (ii) During the problem-solving process, these models fail to effectively integrate their internal knowledge with external edits, making it difficult to address different sub-questions.

A notable example is the baseline model: PokeMQA, which relies heavily on instruction-following capabilities and performs poorly on both LLAMA-3.1-8B and QWEN2.5-3B. This highlights the importance of an effective decomposition mechanism that does not depend on strong instruction follow-up abilities, particularly for smaller models, as it plays a crucial role in overall performance.

Overfitting of parameter-based methods. Our experiments reveal that parameter-based methods perform remarkably well on models with smaller parameter sizes. For example, in the Qwen2.5-3B (1-edited) setting, MEMIT achieves a significantly higher accuracy score of 22.43, compared to 3.83 for MeLLo. This result is unexpected, as prior research suggests that memory-based methods generally exhibit better generalization than parameter-based approaches. To investigate this discrepancy, we conducted a detailed case study and found that MEMIT's high accuracy is primarily driven by model overfitting.

Specifically, after injecting modified knowledge, the model consistently outputs the newly introduced information whenever it encounters related questions, even in contexts where it is not appropriate. The example in Figure 8 provides a detailed explanation of why this phenomenon leads to a higher augmentation bias.

Omission Phenomenon. We also analyze the performance of MeLLo using the original prompts provided with the model implementation. We observe that it leads to omission phenomenon in the decomposition for complex questions, *i.e.,* the MeLLo's decomposition plan skips certain steps, specifically the logical intersection part. Underlying justification in this regard is the fact that the conditional confirmation operations, *e.g.,* logical intersection, does not appear in the multi-hop questions. This showcases that the generalization of decomposition operation through prompt examples is insufficient, highlighting the essence of incorporating examples similar to the question being decomposed. An example illustration in this regard is provided in Appendix Table 8.

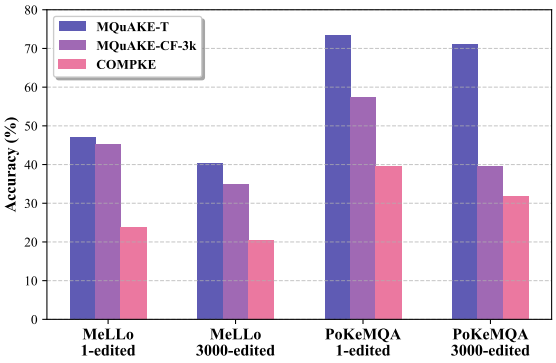


Figure 5: Performance comparison of MeLLo and PoKeMQA on the MQuAKE-T, MQuAKE-CF-3k, and COMPKE datasets on GPT-4O-MINI, with COMPKE presenting more challenging than previous datasets.

Comparision with other Datasets. We select two popular KE datasets (MQuAKE-T and MQuAKE-CF-3k) for comparison with our dataset. Using GPT-4o-mini as the test model, we evaluate the performance of two methods, MeLLo and PoKeMQA, on these datasets (detailed information about MQuAKE and its evaluation metric are shown in appendix D.1 and D.3) and compare them with COMPKE. The results are illustrated in Figure 5. Both methods exhibit lower accuracy on COMPKE than on MQuAKE datasets, indicating that COMPKE presents a greater challenge then previous ones.

6 Conclusion

In this paper, we introduce the concept of complex questions in the context of knowledge editing and propose a new benchmark, COMPKE. Through a comprehensive evaluation of various knowledge editing methods on COMPKE, we find that existing approaches struggle when dealing with complex question scenarios. We analyze the cause of these limitations and suggest that future work can leverage our dataset and evaluation framework to develop more robust and generalizable knowledge editing methods.

Limitations

This work poses following limitations:

- In CompKE, edits are randomly introduced through counterfactual modifications, which may result in discrepancies from actual/real-world modifications.
- The fact triples in CompKE are restricted to one-to-one and one-to-many relations, excluding many-to-many and many-to-one relationships.

Ethics Statement

This work directly deals with updating the capability and/or editing the knowledge of large models. It has the potential for abuse, such as adding poisonous misinformation, malicious content, bias etc. Keeping in view these concerns, we highlight this work must not be used under critical settings.

References

Abhinav Jauhri Abhimanyu Dubey et al. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Muhammad Asif Ali, Nawal Daftardar, Mutayyaba Waheed, Jianbin Qin, and Di Wang. 2024. Mqakeal: Multi-hop question answering under knowledge editing for arabic language.

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-based question answering with knowledge graph. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers*, pages 2503–2514.

Yuchen Cai, Ding Cao, Rongxi Guo, Yaqin Wen, Guiquan Liu, and Enhong Chen. 2024. Editing knowledge representation of language model via rephrased prefix prompts.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2023. Evaluating the ripple effects of knowledge editing in language models.

Fernanda De La Torre, Cathy Mengying Fang, Han Huang, Andrzej Banburski-Fahey, Judith Amores Fernandez, and Jaron Lanier. 2024. Llmr: Real-time prompting of interactive worlds using large language models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–22.

Jingcheng Deng, Zihao Wei, Liang Pang, Hanxing Ding, Huawei Shen, and Xueqi Cheng. 2024. Everything is editable: Extend knowledge editing to unstructured data in large language models.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R Costa-jussà. 2024. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*.

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

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. Model editing harms general abilities of large language models: Regularization to the rescue.

Willis Guo, Armin Toroghi, and Scott Sanner. 2024. Cr-lt-kgqa: A knowledge graph question answering dataset requiring commonsense reasoning and long-tail knowledge. *arXiv preprint arXiv:2403.01395*.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. 2024. Model editing at scale leads to gradual and catastrophic forgetting. *arXiv preprint arXiv:2401.07453*.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2022. Aging with grace: Lifelong model editing with discrete key-value adaptors. *ArXiv*, abs/2211.11031.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2024a. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *Advances in Neural Information Processing Systems*, 36.

Peter Hase, Thomas Hofweber, Xiang Zhou, Elias Stengel-Eskin, and Mohit Bansal. 2024b. Fundamental problems with model editing: How should rational belief revision work in llms? *arXiv preprint arXiv:2406.19354*.

Bing He, Mustaque Ahamad, and Srijan Kumar. 2023. Reinforcement learning-based counter-misinformation response generation: A case study of covid-19 vaccine misinformation. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 2698–2709, New York, NY, USA. Association for Computing Machinery.

Cheng-Hsun Hsueh, Paul Kuo-Ming Huang, Tzu-Han Lin, Che Wei Liao, Hung-Chieh Fang, Chao-Wei Huang, and Yun-Nung Chen. 2024. Editing the mind of giants: An in-depth exploration of pitfalls of knowledge editing in large language models. In

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 9417–9429, Miami, Florida, USA. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Wenyue Hua, Jiang Guo, Mingwen Dong, Henghui Zhu, Patrick Ng, and Zhiguo Wang. 2024. Propagation and pitfalls: Reasoning-based assessment of knowledge editing through counterfactual tasks.

Baixiang Huang, Canyu Chen, Xiongxiao Xu, Ali Payani, and Kai Shu. 2024. Can knowledge editing really correct hallucinations? *arXiv preprint arXiv:2410.16251*.

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

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Jun-Yu Ma, Jia-Chen Gu, Zhen-Hua Ling, Quan Liu, and Cong Liu. 2024. Untying the reversal curse via bidirectional language model editing.

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

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. *ArXiv*, abs/2206.06520.

Kento Nishi, Maya Okawa, Rahul Ramesh, Mikail Khona, Hidenori Tanaka, and Ekdeep Singh Lubana. 2024. Representation shattering in transformers: A synthetic study with knowledge editing.

Jingcheng Niu, Andrew Liu, Zining Zhu, and Gerald Penn. 2024. What does the knowledge neuron thesis have to do with knowledge? *arXiv preprint arXiv:2405.02421*.

Yasumasa Onoe, Michael J. Q. Zhang, Shankar Padmanabhan, Greg Durrett, and Eunsol Choi. 2023. Can lms learn new entities from descriptions? challenges in propagating injected knowledge.

Hao Peng, Xiaozhi Wang, Chunyang Li, Kaisheng Zeng, Jiangshan Duo, Yixin Cao, Lei Hou, and Juanzi Li. 2024. Event-level knowledge editing.

Amit Rozner, Barak Battash, Lior Wolf, and Ofir Lindenbaum. 2024. Knowledge editing in language models via adapted direct preference optimization. *arXiv preprint arXiv:2406.09920*.

Zhengxiang Shi and Aldo Lipani. 2024. Dept: Decomposed prompt tuning for parameter-efficient fine-tuning.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Mengru Wang, Yunzhi Yao, Ziwen Xu, Shuofei Qiao, Shumin Deng, Peng Wang, Xiang Chen, Jia-Chen Gu, Yong Jiang, Pengjun Xie, et al. 2024. Knowledge mechanisms in large language models: A survey and perspective. *arXiv preprint arXiv:2407.15017*.

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bo Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. 2023a. Easyedit: An easy-to-use knowledge editing framework for large language models. *ArXiv*, abs/2308.07269.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, et al. 2023b. Knowledge editing for large language models: A survey. *arXiv preprint arXiv:2310.16218*.

Weixuan Wang, Barry Haddow, and Alexandra Birch. 2023c. Retrieval-augmented multilingual knowledge editing.

Wanli Yang, Fei Sun, Xinyu Ma, Xun Liu, Dawei Yin, and Xueqi Cheng. 2024. The butterfly effect of model editing: Few edits can trigger large language models collapse.

Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. 2024. Knowledge circuits in pretrained transformers. *arXiv preprint arXiv:2405.17969*.

Xunjian Yin, Jin Jiang, Liming Yang, and Xiaojun Wan. 2023. History matters: Temporal knowledge editing in large language model.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. 2024a. Knowledge graph enhanced large language model editing.

Ningyu Zhang, Yunzhi Yao, Bo Tian, Peng Wang, Shumin Deng, Meng Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiao-Jun Zhu, Jun Zhou, and Huajun Chen. 2024b. A comprehensive study of knowledge editing for large language models. *ArXiv*, abs/2401.01286.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023a. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4862–4876, Singapore. Association for Computational Linguistics.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023b. Can we edit factual knowledge by in-context learning? *ArXiv*, abs/2305.12740.

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*.

A Related Work

A.1 More detailed Related Work

Besides benchmarks, many researchers in recent years have explored knowledge editing from various perspectives. There is a type of research that aim to understand the working mechanisms of knowledge editing techniques, such as the relationship between model parameter localization and editing (Wang et al., 2024; Niu et al., 2024; Hase et al., 2024a,b; Ferrando et al., 2024; Gupta et al., 2024; Yao et al., 2024). For example, causal tracing does not effectively indicate the optimal editing location (Hase et al., 2024a), and some researchers have also employed computation graph to uncover the specific impacts on the model's internal behavior of knowledge editing (Yao et al., 2024). Another line of research focuses on enhancing the effectiveness of knowledge editing in specific scenarios (Rozner et al., 2024; Ma et al., 2024; De La Torre et al., 2024; Huang et al., 2024; Deng et al., 2024; Peng et al., 2024; Cai et al., 2024). For instance, bidirectional relationship modeling has been proposed to address consistency issues in bidirectional models (Ma et al., 2024), while real-time knowledge editing methods have been developed to adapt to dynamic environments where knowledge evolves frequently (De La Torre et al., 2024). Additionally, this paper focuses on exploring knowledge editing in the context of complex logical reasoning. Also some studies focus on addressing the side effects of knowledge editing techniques (Hsueh et al., 2024; Gu et al., 2024; He et al., 2023; Hua et al., 2024; Yang et al., 2024; Cohen et al., 2023; Nishi et al., 2024).

A.2 Drawbacks of KGQA Datasets

Although complex questions have been extensively studied in the KG domain, they cannot be directly applied to the knowledge editing field due to two key challenges:

(i) Omission of sub-questions. These datasets do not explicitly provide sub-questions of complex questions. For example, ComplexQuestions(Bao et al., 2016) only includes only the question and its final answer, while ComplexWebQuestions(Talmor and Berant, 2018) provides only a SPARQL statement for each complex question. However, KE requires modifications at the sub-question level. Without explicitly defined sub-questions, introducing targeted edits becomes impractical.

(ii) Knowledge dependency. These data sets do not require models to rely on LLMs' intrinsic knowledge to generate answers, while KE heavily relies on model's internal knowledge. Directly adopting these datasets risks introducing unlearned knowledge into the evaluation, leading to unreliable answers regardless of editing success. In constructing CompKE, we mitigate this by filtering out knowledge instances that cannot be recalled by the model.

B Additional Preliminaries

B.1 Multi-hop Question Answering

A multi-hop question can be represented as $s_1 \xrightarrow{r_1} s_2 \cdots \xrightarrow{r_{n-1}} s_n$, continuously mapping one entity to another. For example. consider the question "Who is the spouse of president of U.S.", it an be represented as U.S. $\xrightarrow{\text{president is}}$ Donald Trump $\xrightarrow{\text{spouse is}}$ Melania Trump.

B.2 Multi-hop Question Answering under KE.

We use $e = (s, r, o \rightarrow o')$ to represent a knowledge edit indicating that the object entity of subject s with relation r is updated from o to o'. This task is to solve multi-hop questions under a batch of knowledge edits $\mathcal{E} = \{e_1, e_2, \cdots\}$.

B.3 MQA with Complex Question Answering.

We consider the previously studied linear multi-hop questions as a special case of complex questions involving continuous mapping of entity through a series of relational links, forming a one-way graph chain: $S_1 \overset{L_1}{\rightarrow} S_2 \overset{L_2}{\rightarrow} \cdots \overset{L_{n-1}}{\rightarrow} S_n$, where n represents the number of reasoning hops. Note that compared to complex questions, here the intermediate set S_i only encompasses a single entity, and L_i only covers one-to-one relation mapping.

C CompKE (Additional Details)

Figure 3 shows the process by which we construct complex question. Figure 10 gives some examples of the structures in CompKE and the corresponding decomposition methods. Table 7 gives the SPARQL which we used to sample facts from WikiData. Table 6 presents the prompt used for converting structured triples into natural lan-

guage. Figure 6 displays the distribution of relation counts across triplets in CompKE.

D Additional Experimental Settings

D.1 MQuAKE

The existing data MQuAKE includes two datasets: MQuAKE-CF-3K, which is based on counterfactual editing, and MQuAKE-T, which is based on real-world changes. These datasets cover k-hop questions ($k \in \{2, 3, 4\}$), each associated with one or more edits. Statistics are presented in Table 4.

Datasets	#Edits	2-hop	3-hop	4-hop	Total
MQuAKE-CF-3K	1	513	356	224	1,093
	2	487	334	246	1,067
	3	-	310	262	572
	4	-	-	268	268
	All	1,000	1,000	1,000	3,000
MQuAKE-T	1	1,421	445	2	1,868

Table 4: Statistics of the MQuAKE dataset.

D.2 Baselines

ROME. ROME by Meng et al. (2022a) uses a locate-then-edit paradigm. For a specific knowledge editing, ROME employs causal tracing to pin-point the exact layer of the MLP module within the Transformer model architecture that encodes the paticular factual association. Then it will perform a rank-one modification on the identified layer.

MEMIT. MEMIT by Meng et al. (2022b) is an evolution of ROME to transcend the inherent limitation that ROME can only edit a single fact at a time. At a time, MEMIT can identify and modify multiple layers in a single pass, allowing for the simultaneous editing of numerous facts.

MeLLo. MeLLo by Zhong et al. (2023) adopts a strategy that alternates between planning and solving stage to solve multi-hop question. It employ a semantic-based retrieval to retrieve relevant edits, and a self-checking mechanism to enable the model to assess the relevance of edits and modifications.

PokeMQA. PokeMQA by Gu et al. (2023) is a memory-based method that extends MeLLo and proposes a two-stage retrieval process to enhance the success rate of retrieving relevant edits.

D.3 Evaluation Metrics

Detailed metrics and mathematical definitions are given below:

(i) Augment Accuracy (Aug) is used to measure whether the edited model can response added knowledge on complex questions. The formula for calculating Aug-Acc is as follows:

$$\mathbb{E}_{q \in \mathcal{Q}}(\left|M'(q) \cap \mathcal{A}_{aug}\right| / |\mathcal{A}_{aug}|) \quad (1)$$

Where $M'(\cdot)$ represents the edited model, and $\mathcal{Q}$ denote the datasets for complex questions, $\mathcal{A}_{aug} = \mathcal{A}' \setminus \mathcal{A}$, $\mathcal{A}'$ is edited answer set and $\mathcal{A}$ is original answer set.

(ii) Retention Accuracy (Ret) is used to measure whether the edited model can retain the original knowledge on complex questions. The formula for calculating Ret-Acc is as follows:

$$\mathbb{E}_{q \in \mathcal{Q}}(\left|M'(q) \cap \mathcal{A}_{ret}\right| / |\mathcal{A}_{ret}|) \quad (2)$$

Where $\mathcal{A}_{ret} = \mathcal{A}' \cap \mathcal{A}$.

(iii) Multi-hop Accuracy (M-Acc) is used to measure the accuracy for multi-hop question under knowledge editing(i.e.,MQuAKE). The formula for calculating M-Acc is as follows:

$$\mathbb{1}\left[\bigvee_{q \in \mathcal{Q}} [M'(q) = a']\right]. \quad (3)$$

Where $M'(\cdot)$ represents the edited model, and $\mathcal{Q}$ and a' denote the multi-hop questions and the final-hop answers for each data, respectively.

D.4 Experiment Setup

Table 5 shows the hyperparameter settings for the parameter-based methods. For the experiments involving ROME and MEMIT, we utilized four NVIDIA Tesla L20 GPUs, with 48GB of memory. A single RTX 4090 GPU was used for MeLLo and PokeMQA.

E Additional Experimental results

E.1 An example for overfitting phenomenon of parameter-based methods.

Figure 8 shows an example of overfitting phenomenon when MEMIT is applied to Qwen2.5-3B.

E.2 Results for Batch Editing(#k-edits)

The results of GPT-3.5-Turbo and GPT-4o-mini for the batch editing, *i.e.,* varying the number of edits (k) are presented in Figure 7.

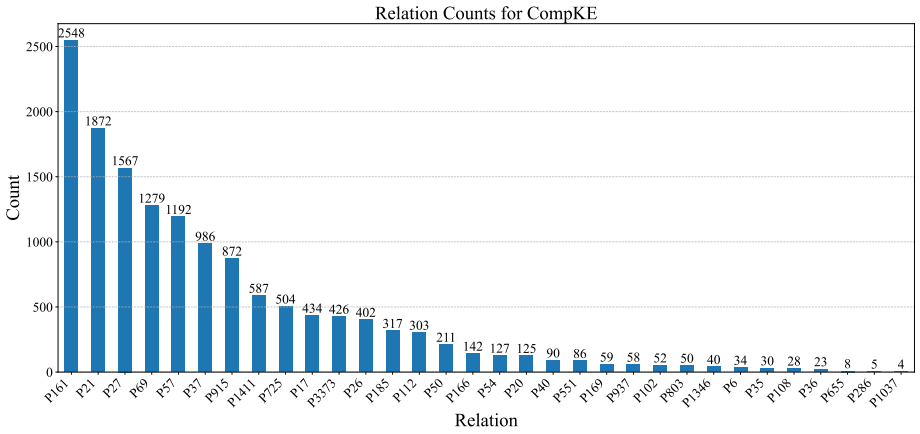


Figure 6: Relations and their frequencies in CompKE

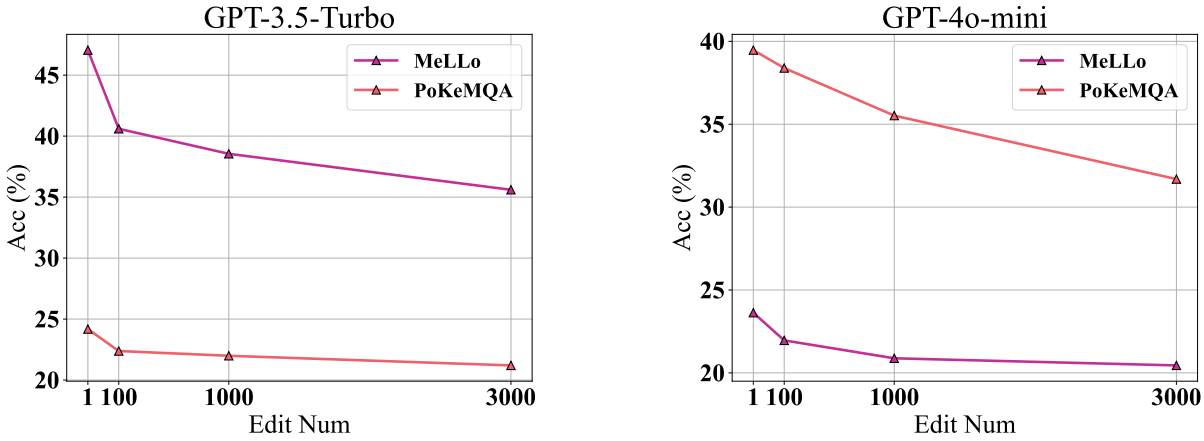


Figure 7: Variation of *Accuracy (Acc)* across GPT-3.5-Turbo and GPT-4o-mini models with varying edit numbers.

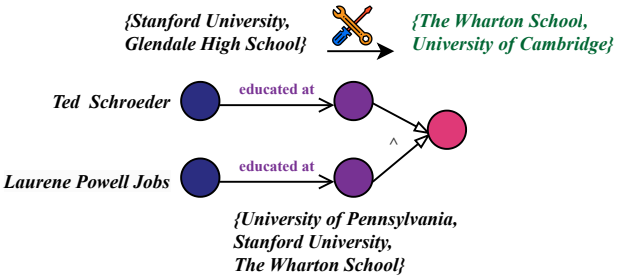
ROME:

```
layers: [5],
fact_token: subject_last,
v_num_grad_steps: 25(for Llama-3.1-8B)||15(for Qwen2.5),
v_lr: 5e-1,
v_loss_layer: 31(for Llama-3.1-8B)||27(for Qwen2.5-7B)||35(for Qwen2.5-3B),
v_weight_decay: 1e-3,
clamp_norm_factor: 4,
kl_factor: 0.0625,
mom2_adjustment: false,
context_template_length_params: [[5, 10], [10, 10]]
```

MEMIT:

```
layers: [3,4,5,6,7,8],
clamp_norm_factor: 4,
layer_selection: all,
fact_token: subject_last,
v_num_grad_steps: 25(for Llama-3.1-8B)||15(for Qwen2.5),
v_lr: 5e-1,
v_loss_layer: 31(for Llama-3.1-8B)||27(for Qwen2.5-7B)||35(for Qwen2.5-3B),
v_weight_decay: 1e-3,
kl_factor: 0.0625,
mom2_adjustment: true,
mom2_update_weight: 15000,
mom2_dataset: wikipedia,
mom2_n_samples: 100000,
mom2_dtype: float32
```

Table 5: Several key hyperparameters for parameter-based KE methods



Question: Which educational institutions did both *Ted Schroeder* and *Laurene Powell Jobs* attend?
Correct Answer Before Editing: Stanford University
Correct Answer After Editing: The Wharton School
Model's Actual Output:
The Wharton School, University of Cambridge

Figure 8: An example of MEMIT applied to Qwen2.5-3B. The correct solution should be for the model to take the **intersection** of the new knowledge about Ted Schroeder's schools ***{The Wharton School, University of Cambridge}*** and Laurene Powell Jobs' schools ***{University of Pennsylvania, Stanford University, The Wharton School}***, yielding the final answer: ***The Wharton School***. However, the model's output is ***{The Wharton School, University of Cambridge}*** (i.e., the newly injected knowledge). While the correct entity *The Wharton School* is correctly included in the final answer(i.e., metric `Aug` is judged as correct), the model after editing merely reiterated the injected knowledge **without** performing the intersection operation. This example illustrates why the `Aug` metric is overestimated.

```
User Input: s1="Electric Dreams",r1="the crew members are",s2="Michael
Madsen",r2="siblings are"
System Output:
Who is someone that is both a member of the cast of the movie Electric Dreams and
a sibling of Michael Madsen?
Of all the cast members of the film Electric Dreams, who are also Michael
Madsen's siblings?
Which members of the cast of the film Electric Dreams are also siblings of
Michael Madsen?

User Input: s1="Felix Sanchez",r1="educated at",s2="Maria Sanchez",r2="educated
at"
System Output:
Which schools have Maria Sanchez and Felix Sanchez both attended?
In the places where Maria Sanchez and Felix Sanchez received their education,
which schools do they have in common?
Which educational institutions did Maria Sanchez and Felix Sanchez both go to?

User Input: s1="Benoit Jacquot",r1="awards that subject was nominated
for",s2="The Public Woman",r2="awards that subject was nominated for"
System Output:
What awards have Benoit Jacquot and "The Public Woman" both been nominated for?
What nominations do Benoit Jacquot and "The Public Woman" share for awards?
What award nominations are shared by Benoit Jacquot and "The Public Woman"?

User Input: s1="Clara Agnelli",r1="siblings are",s2="Giovanni Agnelli",r2="The
founders of the company are"
System Output:
Who are both siblings of Clara Agnelli and founders of the company Giovanni
Agnelli started?
Among Clara Agnelli's siblings, who are also recognized as the founders of the
company by Giovanni Agnelli?
Which individuals are both siblings of Clara Agnelli and acknowledged as the
founders of the company by Giovanni Agnelli?
```

Table 6: A prompt used for converting structured triples into natural language.

SQL Query	Description
`SELECT ?object WHERE { wd:{qid} wdt:pid ?object. FILTER(LANG(?object) = "en"). }`	This SPARQL query retrieves the object associated with the <pid> of entity.
`SELECT (COUNT(?statement) AS ?referencesCount) WHERE { wd:{entity_id} ?p ?statement. ?statement prov:wasDerivedFrom ?source. }`	This SPARQL query retrieves the count of references (i.e., the number of statements that refer to a source) for a specific entity. This query is used to filters out triples with low references counts(i.e.,unpopular entity).
`SELECT ?alias WHERE { wd:{qid} skos:altLabel ?alias. FILTER(LANG(?alias) = "en"). }`	This SPARQL query retrieves the aliases associated with the entity,

Table 7: SPARQL Queries and Descriptions

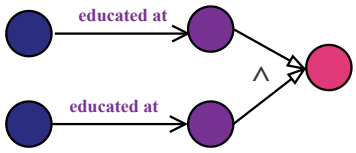
Question: What nominations for awards do both "In America" and Christian Bale share?
Edit Num=1:
Model Output: Both In America and Christian Bale share Golden Globe nominations.
Edit Num=100:
Model Output: Bailmailboxerials屬earable standards, and it is clear that this is a difficult task. Can you give meCloseOperationunded(bind)care(I Lothra 1000... Crimes of the Heart is a crime novel about the heart. The book is a satire of___ I'myncync
Edit Num=1000:
Model Output: 澍澍 greedy澍旄澍blasnown镳ley澍nown" 澍澍澍澍澍nown Cov澍 nick澍 ﷺ真爱igeltotalCount Usa澍放过澍澍放过nown镳bservable finishn澍澍澍不出enkoiment澍放过ocopDAQocop放过澍放` zczeenko Usa澍enko澍放过镳澍onom

Figure 9: When the edit batch size increases, the MEMIT method outputs a large amount of gibberish after models like Qwen-2.5-3B and other smaller models.

Question: **If a person is the director of both Thief and Ali, which country does this person belong to?**
Subquestion: Who is the director of Thief?
Generated answer: The director of Thief is Michael Mann.
Retrieval: The country to which Michael Mann belongs is Italy
Retrieved fact does not contradict to generated answer, so the intermediate answer is: Italy.
Subquestion: Who is the director of Ali?
Generated answer: The director of Ali is Michael Mann.
Retrieval: The country to which Michael Mann belongs is Italy
Retrieved fact does not contradict to generated answer, so the intermediate answer is: Italy.
Final answer: Italy

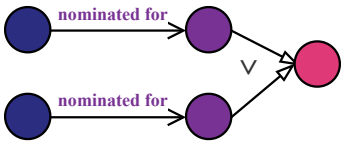
Table 8: The decomposition of a complex question by Mello did not take into account logical operations.

Q: **Which educational institutions did both *Ted Schroeder* and *Laurene Powell Jobs* attend?**



T1: Which educational institution did Ted Schroeder attend?
T2: Which educational institution did Laurene Powell Jobs attend?
T3: Logic Operation: Intersection T1 and T2.

Q: **What awards has either the film *Gladiator* or *Branko Lustig* been nominated for?**

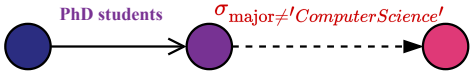


T1: What awards has the film Gladiator been nominated for?
T2: What awards has Branko Lustig been nominated for?
T3: Logic Operation: Union T1 and T2.

Q: **Who among the crew members of *Mortal Kombat: Annihilation* holds American citizenship?**

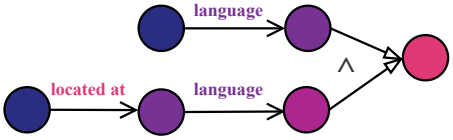
T1:Who are the crew members of the movie Mortal Kombat: Annihilation?
T2:What is the nationality of each person in T1?
T3:Logic Operation: Select persons from T2 whose nationality is American.

Q: **Which of *Nikolaus Joseph von Jacquin's* PhD students did not major in computer science?**



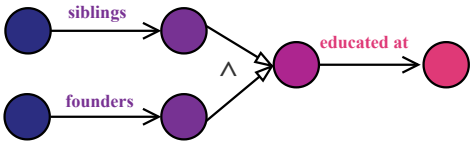
T1: Who are the PhD students of Nikolaus Joseph von Jacquin?
T2: What are the majors of each person in T1?
T3: Logic Operation: Select persons from T2 whose major is not Computer Science.

Q: **Which language spoken in *Palau* is the same as the official language of the country where *Ball State University* is located?**



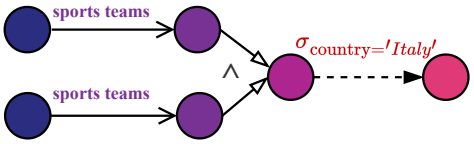
T1: What is the official language of Palau?
T2: What is the location of Ball State University?
T3: What is the official language of T2?
T4: Logic Operation: Intersection T1 and T3.

Q: **If someone is both a sibling of *Mona Simpson* and one of the founders of *Apple*, what is this person's nationality?**



T1: Who are the siblings of Mona Simpson?
T2: Who are the founders of Apple?
T3: Logic Operation: Intersection T1 and T2.
T4: What is the nationality of T3?

Q: **Which sports teams are associated with both *Papin* and *Christophe Dugarry* are located in Italy?**



T1: Which team has Papin been associated with?
T2: Which team has Christophe Dugarry been associated with?
T3: Logic Operation: Intersection T1 and T2.
T4: Where did each team of T3 located?
T5: Logic Operation: Select team from T4 that are located in Italy.

Figure 10: Some typical reasoning structure in CompKE

Relation	Question template	Cloze-style statement template
P40	Who are [S]'s children?	[S]'s children are
P69	Where did [S] receive education?	The university where [S] was educated is
P3373	Who are the siblings of [S]?	[S]'s siblings are
P50	Who are the author(s) of [S]? (list all)	The author(s) of [S] is(are)
P161	Who are the cast members of movie [S]?	The cast members of movie [S] are
P112	Who are the people who founded company [S]?	The people who founded Company [S] are
P54	Which organizations is [S] a member of?	[S] is a member of the following organizations
P915	Where were movie [S] filmed?	The movie [S] was filmed at
P37	What are the official languages of country [S]?	The official languages of country [S] are
P1830	Which companies does S own?	[S] owns the following companies
P6	Who are the heads of government for [S]?	The heads of government for [S] are
P803	What are the professorship ranks for [S]?	The professorship ranks for [S] are
P185	Who are the doctoral students of [S]?	The doctoral students of [S] are
P57	Who is the director of the film [S]?	The film [S] is directed by
P1411	What awards was the film [S] nominated for?	The film [S] is nominated for
P1346	Who are the winners for [S] prize?	The winners for [S] prize are
P286	Who are the head coaches for team [S]?	The head coaches for team [S] are
P166	What awards did [S] receive?	The award received by [S] are
P800	What are the notable works of [S]?	The notable works of [S] are
P725	Who are the voice actors in the movie [S]?	The voice actor in the movie [S] are
P655	Who are the translators of the book [S]?	The translators of the book [S] are
P27	Which country is [S] a citizen of?	The country to which [S] belongs is
P21	What's [S]'s gender?	[S]'s gender is
P169	Who is the CEO of company [S]?	The CEO of company [S] is
P35	Who is the head of state of country [S]?	The head of state of country [S] is
P26	Who is the spouse of [S]?	The spouse of [S] is
P1037	Who is the director of [S]?	The director of [S] is
P20	In which city did [S] die?	[S] died in the city of
P551	Where does [S] live?	[S] lives in the place of
P159	Where is the headquarters of company [S]?	The headquarters of company [S] is located in
P17	In which country is [S] located?	[S] is located in the country of
P108	Who is the employer of [S]?	[S] is an employee in the organization of
P102	Which political party is [S] affiliated with?	[S] is affiliated with the political party of
P937	Where does [S] work?	[S] works in the place of
P140	What is the religion of [S]?	[S] is affiliated with the religion of
P106	What is [S]'s occupation?	[S]'s occupation is
P30	On which continent is country [S] located?	Country [S] is located in the continent of
P38	What is the currency of country [S]?	The currency of country [S] is
P641	Which sport is [S] associated with?	[S] is associated with the sport of
P36	What is the capital of country [S]?	The capital of country [S] is

Table 9: Relations we use to construct CompKE