CompKE: Complex Question Answering under Knowledge Editing

Keyuan Cheng[*,1,3,4], **Zijian Kan**[*,1,4], **Zhixian He**[1,5], **Zhuoran Zhang**[1,3], **Muhammad Asif Ali**[2], **Ke Xu**[4], **Lijie Hu**[†,1,2], **Di Wang**[†,1,2]
[1]Provable Responsible AI and Data Analytics (PRADA) Lab
[2]King Abdullah University of Science and Technology
[3]Peking University [4]South China University of Technology
[5]Sun Yat-sen University

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 conduct an extensive evaluation of four knowledge editing methods on CompKE, revealing that their effectiveness varies notably across different models. For instance, MeLLo attains an accuracy of 39.47 on GPT-4o-mini, but this drops sharply to 3.83 on Qwen2.5-3B. We further investigate the underlying causes of these disparities from both methodological and model-specific perspectives. The datasets are available at https://github.com/kzjkzj666/CompKE.

1 Introduction

Large language models (LLMs) have demonstrated impressive capabilities across a variety of real-world tasks. However, they are still prone to producing outdated, fraudulent, or incorrect information (Wang et al., 2023b; Zhang et al., 2024b; Fang et al., 2023; Yao et al., 2025b,a; Yang et al., 2025; Su et al., 2023b,a). To address this, the field of Knowledge Editing (KE)—which focuses on updating a model's knowledge without costly full-model fine-tuning—has emerged as an active area of research (Wang et al., 2023b; Zhang et al., 2024b).

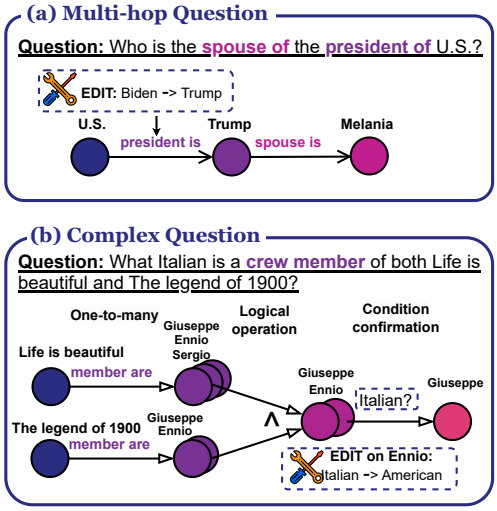


Figure 1: (a) An example of a multi-hop question involving only one-to-one sequential step-by-step reasoning. (b) An example of a complex problem involving one-to-many knowledge mapping, logical operations, and conditional confirmation.

A widely adopted strategy for evaluating the effectiveness of KE methods is to test whether the model can reproduce newly injected knowledge, as seen in benchmarks like ZsRE (Levy et al., 2017) and COUNTERFACT (Meng et al., 2022a). We observe, these benchmarks focus mainly on rote memorization and fail to assess whether the model can effectively utilize the updated knowledge in more complex, contextual scenarios. To address this limitation, MQuAKE (Zhong et al., 2023) introduces multi-hop question answering (MQA) as a more rigorous evaluation, requiring models to integrate and reason over multiple pieces of information. An example in this regard is illustrated in Figure 1, 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, *(b)* determining the president's spouse.

Nevertheless, multi-hop question evaluation re-

[*]Equal Contribution.
[†]Corresponding Author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 2557–2576
July 27 - August 1, 2025

mains limited in scope and does not fully capture a model's ability to flexibly apply newly integrated knowledge. These limitations are evident in three main aspects: *(i) Linear question structure*: The questions typically follow a rigid, sequential pattern, resulting in overly simplistic reasoning chains that can be solved via step by step process. *(ii) One-to-one relations*: Sub-questions are generally based on fact triples with strict one-to-one relationships, which do not reflect the complexity of real-world knowledge. In reality, many facts involve one-to-many relations, such as *"Who are the major shareholders of a company?"*, where a single subject is associated with multiple entities. *(iii) Limited edit operations*: Knowledge edits are mostly restricted to substitutions, neglecting more complex modifications such as additions and deletions that are common in real-world scenarios.

To bridge this gap, we introduce CompKE: **Comp**lex Question Answering under **K**nowledge **E**diting, a new benchmark specifically designed for complex question answering in the context of knowledge editing. Built from Wikidata, CompKE contains 11,924 complex questions. As illustrated in Figure 1(b), CompKE advances beyond existing multi-hop knowledge editing benchmarks in several key ways:
(i) Diverse question structures: Sub-questions in CompKE are flexibly composed to form complex questions, incorporating logical operations, conditional checks, and knowledge mapping.
(ii) One-to-many relations: The underlying fact triples support both one-to-one and one-to-many relationships, better reflecting the complexity of real-world knowledge (e.g., questions involving multiple correct answers).
(iii) Expanded capabilities: CompKE includes a broader range of knowledge edits, systematically covering not only substitutions but also additions and deletions, to more closely mirror real-world knowledge updates.

In order to evaluate the effectiveness of KE methods on CompKE, we conduct an extensive evaluation of leading KE methods on five LLMs spanning diverse model families, encompassing both open-source and closed-source architectures with a range of parameter sizes. Our results reveal that most methods achieve only modest performance on complex question answering tasks. Further analysis across different model scales indicates that parameter-based approaches tend to be more effective for smaller models, while memory-based methods yield better outcomes on larger models with stronger reasoning abilities. 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 an essential area of research for LLMs, allowing them to update their knowledge and remain responsive to new or changing information. To evaluate the effectiveness of KE methods, a range of benchmarks have been developed.

Early benchmarks such as COUNTERFACT (Meng et al., 2022a) focus on counterfactual knowledge updates, while ZsRE (Levy et al., 2017) and MzsRE (Wang et al., 2023c) expand evaluation to zero-shot and multilingual scenarios. ECBD (Onoe et al., 2023) investigates whether newly injected facts can support reasoning over related entities. EasyEdit (Wang et al., 2023a) provides a unified framework for implementing and comparing various state-of-the-art KE approaches. More recent efforts, including MQuAKE (Zhong et al., 2023) and MQA-AEVAL (Ali et al., 2024), extend KE evaluation to multi-hop reasoning tasks. TEMPLAMA (Zheng et al., 2023a) and ATOKE (Yin et al., 2023) address temporal knowledge editing, aiming to update time-sensitive information without interfering with knowledge from other periods.

Despite these advances, existing benchmarks often fail to capture the full complexity of real-world scenarios. In particular, they typically lack support for reasoning over one-to-many relations and for combining entities using logical operations such as intersection and union.

Knowledge Editing Methods. Existing research

on KE can be broadly categorized into parameter-based and memory-based approaches.

Parameter-based methods update a model's internal parameters to encode new or corrected knowledge. Notable examples include ROME (Meng et al., 2022a) and MEMIT (Meng et al., 2022b), which target and modify parameters linked to specific facts, and Transformer-Patcher (Huang et al., 2023), which introduces new neurons to encode edits. To address issues such as high computational cost and catastrophic forgetting, lightweight adaptation techniques like LoRA (Hu et al., 2021), Prompt Tuning (Shi and Lipani, 2024), and QLoRA (Dettmers et al., 2023) have been proposed. Despite their effectiveness for single-fact edits, parameter-based methods often struggle with multi-hop and complex reasoning tasks. Additionally, they are not applicable to closed-source models (e.g., OpenAI GPTs) that are only accessible via APIs, and they generally require more computational resources compared to memory-based approaches.

Memory-based methods maintain an external memory to store knowledge edits, retrieving relevant information at inference time (Cheng et al., 2024b). For example, SERAC (Mitchell et al., 2022) combines semi-parametric editing with retrieval-augmented models, while GRACE (Hartvigsen et al., 2022) leverages adapters and vector matching for knowledge modification. IKE (Zheng et al., 2023b) uses in-context learning with stored demonstrations, and MeLLo (Zhong et al., 2023) stores edited facts externally, incorporating them via prompts. PokeMQA (Gu et al., 2023) introduces a two-stage process for question decomposition and conflict detection, and GLAME (Zhang et al., 2024a) integrates a knowledge graph module to improve retrieval.

In our analysis, we find that MeLLo and PokeMQA are particularly effective for multi-hop reasoning. Consequently, we adopt them as baselines in our experiments to evaluate the generalization of memory-based methods to complex question answering. Additional discussion of related work is provided in Appendix A.

3 Preliminaries

Notations. We represent the knowledge base as a set of triples $\mathcal{D} = \{(s, r, o)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where $\mathcal{E}$ is the set of entities and $\mathcal{R}$ is the set of relations. Each triple (s, r, o) encodes a factual statement indicating that the subject entity s is connected to the object entity o via relation r. To accommodate one-to-many relationships, we generalize the knowledge instance to the form $(s, r, \mathcal{O})$, where $\mathcal{O} = \{o_1, o_2, \ldots\}$ is a set of object entities. For example, (Avatar, actors_are, {Worthington, Saldana, ...}) captures that the movie Avatar has multiple actors.

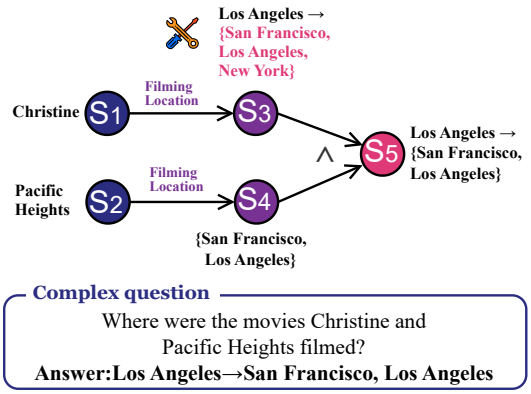


Figure 2: An example of a complex question under knowledge editing. Knowledge editing occurs in the first sub-question, where the filming location of *Christime* is modified from { Los Angeles} to {San Francisco, Los Angeles, New York}.

3.1 Complex Questions

Building on the example introduced earlier, we now formally define the notion of complex questions considered in this work. For a brief overview of multi-hop question answering (MQA) and MQA in the context of knowledge editing (KE), please refer to Appendix B.1.

We represent a complex question Q as a graph-structured reasoning problem, that is, $Q = (\mathbf{S}, \mathbf{L})$, where $\mathbf{S} = \{S_1, S_2, \ldots\}$ is a collection of *intermediate entity sets*, and $\mathbf{L} = \{L_1, L_2, \ldots\}$ is a collection of *reasoning links*. Each $S_i \in \mathbf{S}$ is itself a set of entities, i.e., $S_i = \{s_1, s_2, \ldots\}$, which allows us to naturally capture both one-to-one and one-to-many knowledge relations. Each $L_i \in \mathbf{L}$ denotes a reasoning link. Unlike standard relations in knowledge graphs—which simply connect one entity s_i to another s_j via a relation r—our reasoning links generalize this notion to support richer operations, including conditional confirmation and logical operations, as detailed below.

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

***(a)* Knowledge-related Links:** These links enable traversal between sets of entities, allowing the

reasoning process to progress from one set $S_i \in \mathbf{S}$ to another set $S_j \in \mathbf{S}$ based on the underlying knowledge base. We further distinguish the following types:

(i) Knowledge Mapping. Given a set S_i, a knowledge mapping link connects S_i to the set of adjacent entities $S_j = \bigcup_{s \in S_i} A_r(s)$, where $A_r(s) = \{s' \mid (s, r, s') \in \mathcal{D}\}$ denotes all entities s' related to s via relation r.

(ii) Condition Confirmation. Given a relation r and a target entity s', this link selects the subset of entities from S_i that satisfy the condition of being connected to s' via r. Formally, $S_j = \{s \in S_i \mid (s, r, s') \in \mathcal{D}\}$, i.e., all s in S_i for which the triple (s, r, s') exists in the knowledge base. This operation checks whether each s in S_i is related to s' through r.

***(b)* Logical Links:** Logical links enable the application of set-based logical operations over collections of intermediate entity sets $\{S_1, S_2, \ldots, S_n\} \subseteq \mathbf{S}$. These operations facilitate more expressive reasoning by combining or filtering entities across multiple sets. The primary logical operations considered are:

(i) Intersection. The intersection operation returns the set of entities that are present in all input sets, formally defined as $S_j = \bigcap_{k=1}^{n} S_k$.

(ii) Union. The union operation aggregates all entities that appear in any of the input sets, given by $S_j = \bigcup_{k=1}^{n} S_k$.

Example 1. Figure 2 provides an illustrative example of a complex question involving multiple reasoning links: *"Where were the movies Christine and Pacific Heights filmed?"*. The intermediate entity sets are as follows: $S_1 = \{\text{Christine}\}$, $S_2 = \{\text{Pacific Heights}\}$, $S_3 = \{\text{Los Angeles}\}$, $S_4 = \{\text{San Francisco}, \text{Los Angeles}\}$, and $S_5 = \{\text{Los Angeles}\}$. The reasoning proceeds in three steps: (1) L_1: map S_1 to S_3 via the `filming_at` relation; (2) L_2: map S_2 to S_4 via the same relation; (3) L_3: apply a logical intersection between S_3 and S_4 ($S_3 \cap S_4$) to yield the final answer, $S_5 = \{\text{Los Angeles}\}$.

Complex Question Answering under Knowledge Editing. We formalize a knowledge edit as $e = (s, r, \mathcal{O} \rightarrow \mathcal{O}')$, where the object set $\mathcal{O}$ associated with subject s and relation r is updated to a new set $\mathcal{O}'$, supporting one-to-many modifications. The model is assumed to have access to the original knowledge base $\mathcal{D}$. Given a set of edits $\mathcal{E} = \{e_1, e_2, \ldots\}$, we define the set of knowledge to be removed as $\mathcal{D}^{\mathcal{E}}_{del} = \{(s_i, r_i, \mathcal{O}_i) \mid e_i \in \mathcal{E}\}$ and the set of knowledge to be added as $\mathcal{D}^{\mathcal{E}}_{add} = \{(s_i, r_i, \mathcal{O}'_i) \mid e_i \in \mathcal{E}\}$. The updated knowledge base is then given by: $\mathcal{D}' = (\mathcal{D} - \mathcal{D}^{\mathcal{E}}_{del}) \cup \mathcal{D}^{\mathcal{E}}_{add}$. The objective is that, following the application of edits and the resulting update of the knowledge base to $\mathcal{D}'$, the LLM is able to correctly answer the complex question Q by utilizing the modified knowledge.

4 CompKE

While complex questions frequently arise in real-world scenarios, they are insufficiently addressed in LLM-based question answering within the context of knowledge editing. Existing benchmarks mainly emphasize linear multi-hop questions, which constrains their ability to assess more intricate queries. To address this limitation, we introduce CompKE: **Compl**ex Question Answering under **K**nowledge **E**diting. Below, we present an overview of CompKE and outline the key steps of its construction process.

4.1 Dataset Construction

Overview. The workflow of data construction process, illustrated in Figure 3, consists of six main steps. We begin by extracting factual triples from Wikidata. In the second step, we select relevant relations and sample corresponding triples. Third, we assemble these triples into complex questions featuring diverse reasoning structures, introducing edits at suitable points within the questions. Step four involves applying counterfactual modifications, followed by step five, where we filter out conflicting instances to ensure consistency. In the final step, the structured questions are transformed into natural language. Each step is described in more detail below.

Step 1: Collecting Relation Templates. We begin by selecting 37 relations from Wikidata's *List of Properties* using a two-stage approach. First, we focus on one-to-many relations (such as family-child, book-authors, and movie-actors), which are crucial for mapping knowledge that connects a single entity to multiple others. Second, we add one-to-one relations (such as country-capital and person-hometown) that represent core attributes of entities, supporting both direct mapping and conditional reasoning. To ensure the dataset's real-world relevance, we give preference to relations that are frequently encountered in ev-

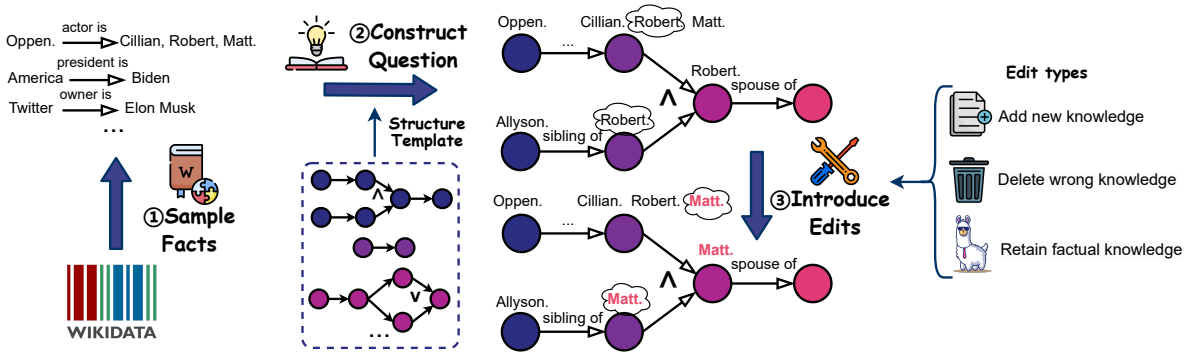


Figure 3: The construction process of CompKE.

eryday contexts. The complete set of relation templates used in CompKE is listed in Appendix Table 9.

Step 2: Sampling Facts. After selecting the relation templates, we construct the knowledge base $\mathcal{D}$, prioritizing widely recognized facts over obscure ones. We sample single-hop knowledge triples from Wikidata based on the chosen relation templates and rank them according to their frequency of access, giving precedence to the most commonly referenced triples. To further ensure the relevance and recallability of the knowledge, we utilize GPT-3.5-Turbo-instruct to filter out facts that the model cannot recall. The resulting knowledge base $\mathcal{D}$ forms the foundation for generating complex questions.

Step 3: Constructing Complex Questions. Complex questions typically exhibit structured reasoning patterns, as illustrated in Figure 2, where knowledge mapping is often combined with logical operations such as intersection. To systematically capture these reasoning structures, we begin by manually curating a set of high-quality complex questions to serve as seed examples. From these, we abstract their underlying reasoning structures by removing intermediate entities, resulting in reusable templates. These templates are then instantiated with real-world facts from $\mathcal{D}$ to generate concrete complex questions. The instantiation process starts by randomly selecting the leaf nodes, and then iteratively determining the intermediate entities through logical operations or by referencing knowledge in $\mathcal{D}$, repeating this process until all entities in the reasoning structure are specified.

To maintain the quality and relevance of the generated questions, we apply several filtering criteria during instantiation: *(i)* questions that lack a valid answer, *(ii)* questions where the set of intermediate entities is empty, and *(iii)* questions in which entities participating in logical operations are of incompatible types. Representative examples of relational structures and their instantiated complex questions are shown in the Appendix (Figure 10).

Step 4: Introducing Counterfactual Edits. To simulate real-world knowledge updates, we introduce counterfactual edits into the knowledge base. For each complex question, we randomly select a knowledge mapping of the form $(s, r, \mathcal{O})$ and apply an edit $e = (s, r, \mathcal{O}')$. In contrast to prior benchmarks that focus solely on one-to-one relations and simple entity substitutions, our approach supports edits on one-to-many relations, resulting in more intricate modifications. To systematically capture the nature of these changes, we define three fundamental operations—addition, deletion, and retention—which can be combined to represent any edit:
(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. We provide an example of a counterfactual edit involving the management of "Microsoft": (Microsoft, managers_are, {John, Smith, Dave} $\rightarrow$ {Smith, Eden, Keyes}). In this case, the edit deletes {John}, retains {Smith}, and adds {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 $e_j =$

$\mathcal{E}$	(*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.

$(s_j, r_j, \mathcal{O}_j \rightarrow \mathcal{O}_j^*)$, with $s_i = s_j$ and $r_i = r_j$, but $\mathcal{O}_i^* \neq \mathcal{O}_j^*$. This implies that conflicting facts may exist within the same batch, which can undermine the validity of the evaluation if introduced together. To address this, we detect and group all conflicting cases, and then randomly retain only one instance from each group.

Step 6: Phrasing in Natural Language. Building on steps 1–5, we generate complex questions involving edits, where each question consists of multiple fact triples. To enable evaluation by LLMs, these structured questions are converted into natural language. Specifically, for each reasoning structure defined in Step 3, we manually curate eight high-quality examples. Using GPT-4o-mini, we then generate three natural language variants for each structured question. Additional details on construction can be found in Appendix C.

4.2 Dataset Summary

Table 2 summarizes the distribution of our dataset along two key axes: `Edit_num` and `Step_num`. Here, `Edit_num` indicates how many triples are edited within each complex question. The majority of questions in CompKE involve a single edit, while a smaller proportion feature two or more edits. `Step_num` captures the number of reasoning steps required to solve each question. Most questions require three reasoning steps, with four-step and five-step questions appearing less frequently. This distribution highlights the predominance of moderately complex questions in our dataset, while still providing a range of multi-step and multi-edit scenarios for comprehensive evaluation.

#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.

Example 3. Table 1 provides a representative example from CompKE, showcasing a complex question formed by merging two sub-questions using an intersection operation. In this example, the edit occurs in the first sub-question, where Christine's filming locations are modified from {Los Angeles} to {San Francisco, Los Angeles, New York}. As a result, San Francisco is included in the final answer.

5 Experiments

In this section, we present a thorough evaluation of state-of-the-art knowledge editing methods on CompKE. Our analysis focuses on three key aspects: the ability to recall newly added knowledge, the retention of existing knowledge, and overall accuracy. We further investigate how the performance of these methods varies with increasing edit batch sizes (*i.e.*, the number of edits applied simultaneously). Through detailed case studies, we identify several noteworthy phenomena, such as overfitting in parameter-based approaches, model collapse as batch size grows, and the omission phenomenon observed in memory-based methods.

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

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.

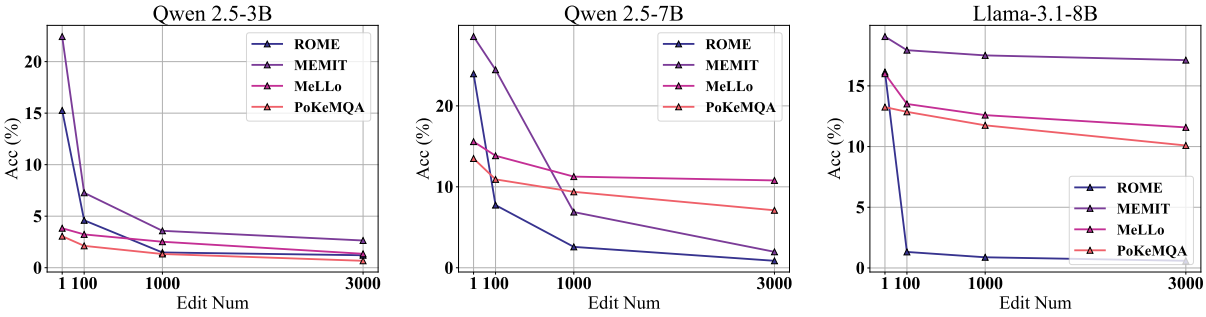


Figure 4: Variation of *Accuracy* (Acc) across QWEN2.5-3B, QWEN2.5-7B, and LLAMA-3.1-8B models with varying edit numbers. Results for GPT-3.5-TURBO and GPT-4O-MINI are provided in Appendix E.2.

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): Measures the number of newly introduced entities that are correctly added to the answer list after the knowledge edit, relative to the original list.

(ii) Retain Accuracy (Ret): Quantifies the number of entities that remain present in both the original and edited answer lists, reflecting the model's ability to retain unaltered knowledge.

(iii) Accuracy (Acc): Calculated as the average of Aug and Ret, this metric provides an overall assessment of the model's capability to answer complex questions following knowledge editing. Detailed description and mathematical formulations of these evaluation metrics are provided in Appendix D.3.

Example 4. As illustrated in Figure 2, suppose the original answer is {Los Angeles}, and after editing, it becomes {San Francisco, Los Angeles}. The Aug metric checks if the model successfully adds the new entity ({San Francisco}) to its answer, while the Ret metric evaluates whether the model preserves the existing entity ({Los Angeles}) across both versions. The overall Acc score, computed as the average of Aug and Ret, reflects how well the model incorporates new knowledge without losing previously acquired 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

Table 3 presents the main experimental results. Overall, MeLLo attains the best performance in the 1-edit scenario on GPT-3.5-Turbo, achieving an `Aug` score of 49.21. Comparing the different approaches, we find that memory-based methods underperform on smaller models (e.g., QWEN2.5-3B), likely due to their dependence on strong instruction-following and reasoning abilities. Conversely, parameter-based methods are more suitable for smaller models, but their performance drops sharply as the number of edits in a batch increases. Below, we analyze these trends in greater detail.

Batch Editing (#k-edits). Figure 4 shows how the accuracy of the four methods changes on QWEN2.5-3B, QWEN2.5-7B, and LLAMA-3.1-8B as the number of edits increases. Additional results for GPT-3.5-TURBO and GPT-4O-MINI are provided in Appendix E.2.

Our findings indicate that memory-based methods exhibit a gradual decrease in performance as the edit batch size (k) grows. In contrast, parameter-based methods experience a much steeper decline, particularly when the number of edits surpasses a certain threshold. Notably, for $k \geq 100$, these models often lose coherence, resulting in inconsistent answers and irrelevant outputs, as further illustrated in Appendix Figure 9.

Performance on Smaller Models. We observe that for smaller language models, such as QWEN2.5-3B, memory-based knowledge editing methods underperform compared to parameter-based approaches. This performance gap can be explained by two main reasons:

(i) Limited Instruction-Following Ability. Smaller models often lack the advanced instruction-following and reasoning skills required to interpret and execute complex prompts, especially those involving multi-step response planning or decomposition. As a result, when memory-based methods rely on the model to follow detailed instructions or structured plans, these models frequently fail to generate answers in the expected format or to complete all necessary reasoning steps.

(ii) Difficulty Integrating Edited Knowledge. In the process of answering complex questions, smaller models struggle to effectively combine their internal knowledge with newly injected information from external edits. This makes it challenging for them to address sub-questions that require synthesizing both original and updated knowledge, leading to incomplete or incorrect answers.

A concrete example of this limitation is seen with the PokeMQA baseline, which depends heavily on the model's instruction-following capabilities. PokeMQA exhibits poor performance not only on QWEN2.5-3B but also on larger models like LLAMA-3.1-8B when those models' instruction-following is insufficient. This underscores the need for decomposition mechanisms that are robust to weaker instruction-following, especially in smaller models, as such mechanisms are critical for achieving strong performance in knowledge editing tasks.

Overfitting in Parameter-Based Methods. Interestingly, our experiments show that parameter-based methods can achieve surprisingly high accuracy on smaller models. For instance, in the Qwen2.5-3B (1-edit) scenario, MEMIT attains an accuracy of 22.43, far surpassing MeLLo's 3.83. This result is counterintuitive, as previous studies generally find that memory-based methods generalize better than parameter-based ones.

To better understand this phenomenon, we conducted a detailed case analysis and discovered that the high accuracy of parameter-based methods like MEMIT is largely due to overfitting. After the model's parameters are updated with new knowledge, the model tends to overproduce the newly injected information, outputting it in response to any related question—even when it is not contextually appropriate. This behavior artificially inflates the augmentation metric, as the model appears to recall the new knowledge very well, but in reality, it is simply repeating the edited content indiscriminately. Figure 8 illustrates how this overfitting leads to a higher augmentation score, highlighting a key limitation of parameter-based editing approaches on smaller models.

Omission Phenomenon. We further investigate the performance of MeLLo by evaluating it with the original prompt templates provided in its official implementation. Our analysis reveals a notable issue: when decomposing complex questions, MeLLo's generated plans sometimes omit critical reasoning steps—most notably, the logical intersection step that is essential for correctly answering multi-hop questions.

This omission occurs because the original prompt examples used to guide MeLLo's decomposition do not include cases that require condi-

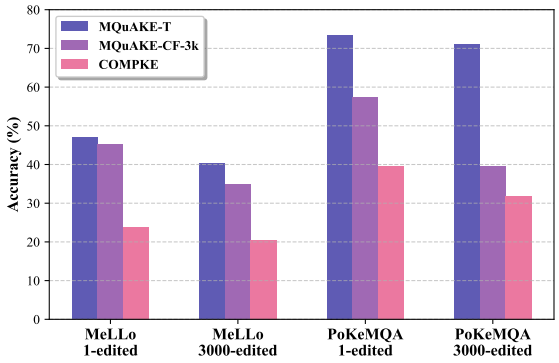


Figure 5: Performance comparison of MeLLo and PoKeMQA on the MQuAKE-T, MQuAKE-CF-3k, and CompKE datasets on GPT-4o-mini, with CompKE presenting more challenging than previous datasets.

tional confirmation operations, such as logical intersections. As a result, MeLLo fails to generalize to questions that demand these reasoning patterns, and its decomposition plan skips necessary steps.

This finding highlights a key limitation: the effectiveness of decomposition-based methods like MeLLo heavily depends on the diversity and representativeness of the prompt examples. If the prompt examples do not cover the full range of reasoning operations needed for complex questions, the model is likely to miss important steps during decomposition. Therefore, it is crucial to include prompt examples that closely resemble the structure and logic of the target questions to ensure robust generalization.

A concrete example illustrating this omission phenomenon is provided in Appendix Table 8.

Comparision with other Datasets. To assess the relative difficulty of our benchmark, we compare it with two widely used knowledge editing datasets: MQuAKE-T and MQuAKE-CF-3k. We evaluate the performance of MeLLo and PoKeMQA on all three datasets using GPT-4o-mini as the test model. Details about the MQuAKE datasets and their evaluation metrics are provided in Appendix D.1 and D.3. As shown in Figure 5, both methods achieve noticeably lower accuracy on CompKE compared to the MQuAKE datasets. This result suggests that CompKE is more challenging and better suited for evaluating the robustness of knowledge editing methods on complex questions.

6 Conclusion

In this paper, we formalize complex questions in knowledge editing—those requiring multi-step reasoning, logical composition, or integrating new and existing knowledge. To rigorously evaluate this challenging setting, we introduce CompKE, a benchmark designed to test current knowledge editing methods on such questions.

Our experiments show that both parameter-based and memory-based approaches struggle with complex questions, often losing answer accuracy due to overfitting or difficulty with multi-step reasoning, especially in smaller models. We analyze these failure modes, highlighting issues like limited instruction-following, challenges in integrating edits, and missing reasoning steps in decomposition-based methods.

By releasing CompKE and our evaluation framework, we aim to spur the development of more robust knowledge editing techniques. We hope future work will build on our findings to create methods that better handle the demands of complex question answering, improving the reliability of knowledge editing in large language models.

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 that this work must not be used under critical settings.

Acknowledgements

This work is supported in part by the funding BAS/1/1689-01-01, URF/1/4663-01-01, REI/1/5232-01-01, REI/1/5332-01-01, and URF/1/5508-01-01 from KAUST, and funding from KAUST - Center of Excellence for Generative AI, under award number 5940.

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.

Muhammad Asif Ali, Yifang Sun, Bing Li, and Wei Wang. 2020. Fine-grained named entity typing over distantly supervised data based on refined representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34(05), pages 7391–7398.

Muhammad Asif Ali, Yifang Sun, Bing Li, and Wei Wang. 2021. Fine-grained named entity typing over distantly supervised data via refinement in hyperbolic space. *CoRR*.

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.

Keyuan Cheng, Muhammad Asif Ali, Shu Yang, Gang Lin, Yuxuan Zhai, Haoyang Fei, Ke Xu, Lu Yu, Lijie Hu, and Di Wang. 2024a. Leveraging logical rules in knowledge editing: A cherry on the top. *arXiv preprint arXiv:2405.15452*.

Keyuan Cheng, Gang Lin, Haoyang Fei, Lu Yu, Muhammad Asif Ali, Lijie Hu, Di Wang, et al. 2024b. Multi-hop question answering under temporal knowledge editing. *arXiv preprint arXiv:2404.00492*.

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.

Xiang Fang, Daizong Liu, Wanlong Fang, Pan Zhou, Yu Cheng, Keke Tang, and Kai Zou. 2023. Annotations are not all you need: A cross-modal knowledge transfer network for unsupervised temporal sentence grounding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8721–8733.

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.

Yihuai Hong, Yuelin Zou, Lijie Hu, Ziqian Zeng, Di Wang, and Haiqin Yang. 2024. Dissecting fine-tuning unlearning in large language models. *arXiv preprint arXiv:2410.06606*.

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.

Lijie Hu, Liang Liu, Shu Yang, Xin Chen, Hongru Xiao, Mengdi Li, Pan Zhou, Muhammad Asif Ali, and Di Wang. 2024. A hopfieldian view-based interpretation for chain-of-thought reasoning. *arXiv preprint arXiv:2406.12255*.

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.

Jinyan Su, Terry Yue Zhuo, Jonibek Mansurov, Di Wang, and Preslav Nakov. 2023a. Fake news detectors are biased against texts generated by large language models. *arXiv preprint arXiv:2309.08674*.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023b. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540*.

Yi Su, Jiayi Zhang, Shu Yang, Xinhai Wang, Lijie Hu, and Di Wang. 2025. Understanding how value neurons shape the generation of specified values in llms. *arXiv preprint arXiv:2505.17712*.

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.

Shu Yang, Shenzhe Zhu, Ruoxuan Bao, Liang Liu, Yu Cheng, Lijie Hu, Mengdi Li, and Di Wang. 2024a. What makes your model a low-empathy or warmth person: Exploring the origins of personality in llms. *arXiv preprint arXiv:2410.10863*.

Shu Yang, Shenzhe Zhu, Zeyu Wu, Keyu Wang, Junchi Yao, Junchao Wu, Lijie Hu, Mengdi Li, Derek F Wong, and Di Wang. 2025. Fraud-r1: A multi-round benchmark for assessing the robustness of llm against augmented fraud and phishing inducements. *arXiv preprint arXiv:2502.12904*.

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

Junchi Yao, Jianhua Xu, Tianyu Xin, Ziyi Wang, Shenzhe Zhu, Shu Yang, and Di Wang. 2025a. Is your llm-based multi-agent a reliable real-world planner? exploring fraud detection in travel planning. *arXiv preprint arXiv:2505.16557*.

Junchi Yao, Shu Yang, Jianhua Xu, Lijie Hu, Mengdi Li, and Di Wang. 2025b. Understanding the repeat curse in large language models from a feature perspective. *arXiv preprint arXiv:2504.14218*.

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.

Lin Zhang, Wenshuo Dong, Zhuoran Zhang, Shu Yang, Lijie Hu, Ninghao Liu, Pan Zhou, and Di Wang. 2025a. Eap-gp: Mitigating saturation effect in gradient-based automated circuit identification. *arXiv preprint arXiv:2502.06852*.

Lin Zhang, Lijie Hu, and Di Wang. 2025b. Mechanistic unveiling of transformer circuits: Self-influence as a key to model reasoning. *arXiv preprint arXiv:2502.09022*.

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(1).

Zhuoran Zhang, Yongxiang Li, Zijian Kan, Keyuan Cheng, Lijie Hu, and Di Wang. 2024c. Locate-then-edit for multi-hop factual recall under knowledge editing. *arXiv preprint arXiv:2410.06331*.

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 Additional Related Work

In addition to the development of benchmarks, recent years have seen a surge of research into knowledge editing from multiple angles. One major research direction seeks to uncover the underlying mechanisms of knowledge editing methods via mechanistic interpretability (Zhang et al., 2025a; Hong et al., 2024; Yang et al., 2024a; Zhang et al., 2025b; Hu et al., 2024; Su et al., 2025). For example, several works investigate how knowledge is localized within model parameters and how edits propagate through the network (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; Zhang et al., 2024c; Cheng et al., 2024a). Notably, Hase et al. (2024a) demonstrate that causal tracing—a technique often used to identify where to intervene in a model—does not always reliably pinpoint the optimal location for editing. Other studies leverage computation graphs to analyze how knowledge edits affect the internal computations and representations of models (Yao et al., 2024).

A second line of research aims to improve the effectiveness of knowledge editing in specific contexts or applications (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 introduced to address consistency issues that arise when editing knowledge in models that must reason about relationships in both directions (Ma et al., 2024). Additionally, real-time knowledge editing techniques have been proposed to enable models to adapt quickly in dynamic environments where facts may change frequently (De La Torre et al., 2024).

This paper specifically investigates knowledge editing in the context of complex logical reasoning, an area that remains underexplored. Furthermore, another important research focus is on understanding and mitigating the side effects of knowledge editing. Editing a model's knowledge can inadvertently impact unrelated facts or reasoning abilities, a phenomenon documented in several recent works (Hsueh et al., 2024; Gu et al., 2024; He et al., 2023; Hua et al., 2024; Yang et al., 2024b; Cohen et al., 2023; Nishi et al., 2024). These studies highlight the need for careful evaluation of both the intended and unintended consequences of knowledge edits.

A.2 Knowledge Graph Question Answering.

Several complex question answering datasets have been developed in the Knowledge Graph (KG) domain, inspired by KGs' ability to store entity-specific information (Ali et al., 2020, 2021). For example, ComplexQuestions (Bao et al., 2016) assesses the ability of KG-based systems to answer queries involving multiple constraints. MetaQA (Zhang et al., 2018) is a multi-hop dataset in the movie domain that includes both textual and audio modalities, requiring reasoning over up to three hops. ComplexWebQuestions (Talmor and Berant, 2018), constructed on the Freebase knowledge base, features complex questions that require aggregating information from multiple web sources. CR-LT-KGQA (Guo et al., 2024) targets commonsense reasoning and long-tail knowledge.

While complex question answering has been widely explored in the knowledge graph (KG) community, existing KGQA datasets are not directly suitable for evaluating knowledge editing (KE) methods. This is primarily due to two fundamental limitations:

(i) Lack of explicit sub-question decomposition. Most KGQA datasets do not provide the intermediate sub-questions that compose a complex question. For instance, the ComplexQuestions dataset (Bao et al., 2016) contains only the overall question and its final answer, omitting any breakdown into simpler reasoning steps. Similarly, ComplexWebQuestions (Talmor and Berant, 2018) offers only a SPARQL query for each question, which encodes the reasoning path but does not explicitly enumerate the sub-questions. In the context of knowledge editing, it is often necessary to target and modify specific sub-components of a reasoning chain. Without clearly defined sub-questions, it becomes infeasible to perform or evaluate fine-grained edits, as there is no direct mapping between edits and the reasoning steps they affect.

(ii) Insufficient reliance on model-internal knowledge. Another key issue is that KGQA datasets typically assume access to an external knowledge base (the KG itself) for answering questions. As a result, models can answer questions by retrieving facts from the KG, rather than relying on their own parametric (internal) knowledge. In contrast,

knowledge editing research focuses on modifying and evaluating the information stored within the model itself. If a dataset requires knowledge that the model has not already learned, or that is not present in its parameters, then editing operations and their evaluation become unreliable: the model may fail to answer correctly regardless of whether the edit was successful. To address this, when constructing CompKE, we carefully filter out any knowledge instances that the model cannot already recall, ensuring that all evaluated edits pertain to knowledge the model actually possesses.

In summary, the absence of explicit sub-question structure and the lack of dependence on model-internal knowledge make standard KGQA datasets ill-suited for knowledge editing research. Our dataset construction process is designed to overcome these challenges.

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 language. 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

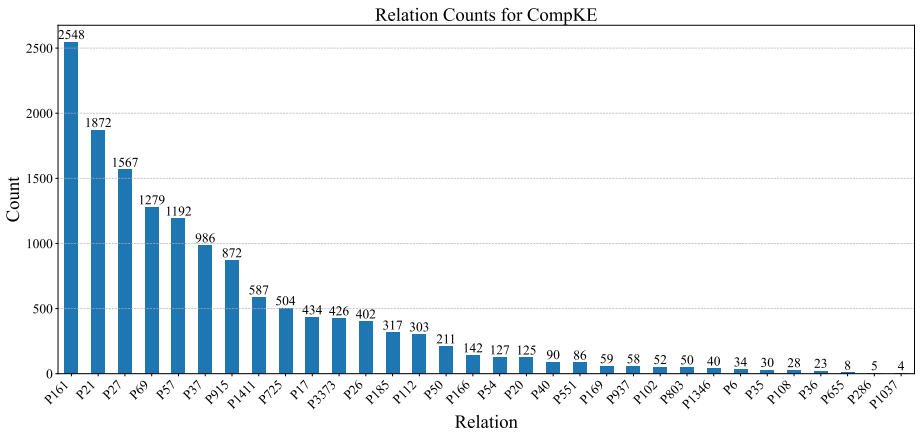


Figure 6: Relations and their frequencies in CompKE

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.

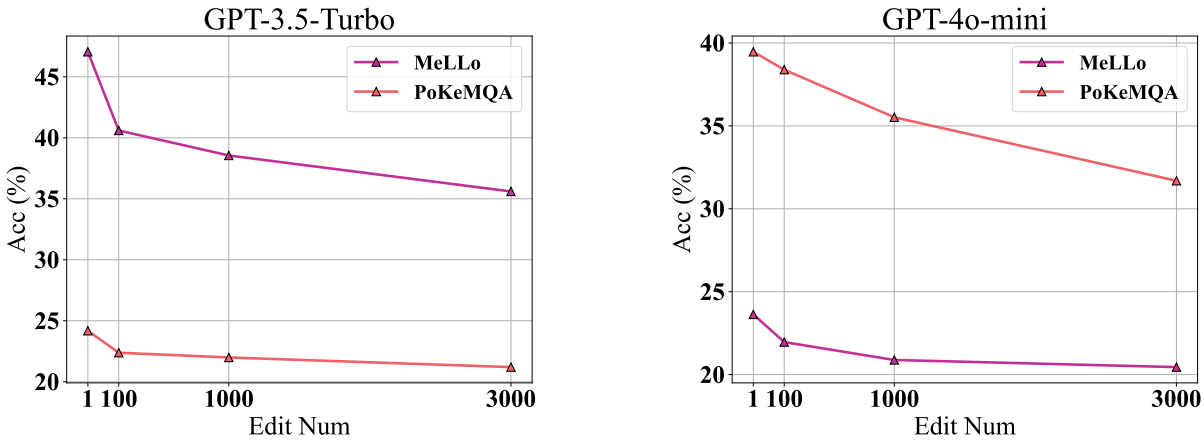
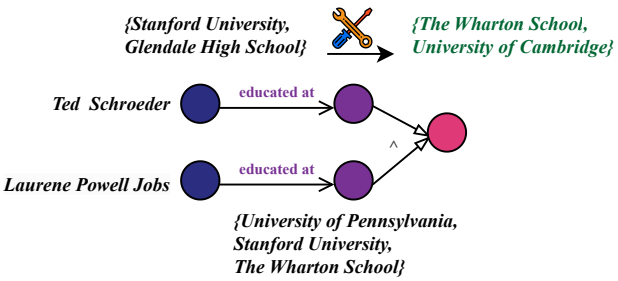


Figure 7: Variation of *Accuracy (Acc)* across GPT-3.5-Turbo and GPT-4o-mini models with varying edit numbers.



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.

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nown Cov澍 nick澍 ﷺ真爱igeltotalCount Usa澍放过澍澍放过nown镳bservable finish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.

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

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).

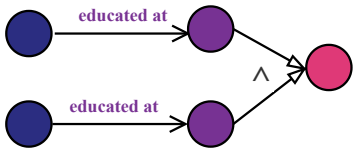
SELECT ?alias WHERE {
wd:{qid} skos:altLabel ?alias.
FILTER(LANG(?alias) = "en").
}
	This SPARQL query retrieves the aliases associated with the entity,

Table 7: SPARQL Queries and Descriptions

Question: **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

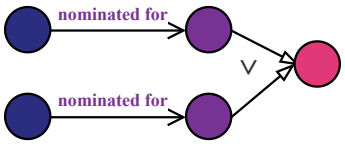
Table 8: The decomposition of a complex question by Mello did not take into account logical operations.

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?**

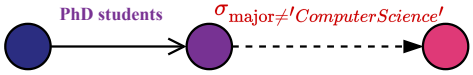


T1: What awards has the film Gladiator been nominated for?
T2: What awards has Branko Lustig been nominated for?
T3: Logic Operation: Union T1 and T2.

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

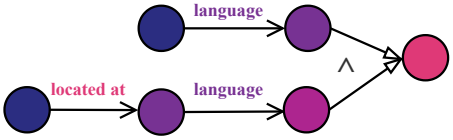
T1:Who are the crew members of the movie Mortal Kombat: Annihilation?
T2:What is the nationality of each person in T1?
T3:Logic Operation: Select persons from T2 whose nationality is American.

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



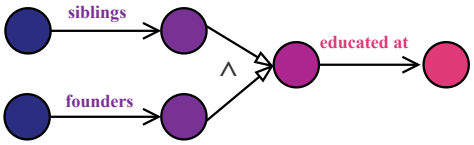
T1: Who are the PhD students of Nikolaus Joseph von Jacquin?
T2: What are the majors of each person in T1?
T3: Logic Operation: Select persons from T2 whose major is not Computer Science.

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



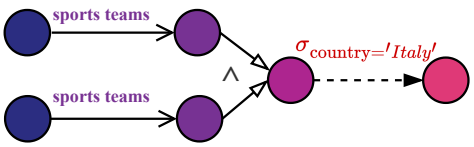
T1: What is the official language of Palau?
T2: What is the location of Ball State University?
T3: What is the official language of T2?
T4: Logic Operation: Intersection T1 and T3.

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