CompKE: Complex Question Answering under Knowledge Editing

Anonymous ACL submission

Abstract

Knowledge Editing—Efficiently modifying the knowledge in large language models has gathered a great attention. Current benchmarks primarily use multi-hop question answering to assess and analyze the newly injected/updated knowledge. We argue, these benchmarks fall short of evaluating how effectively the updated model applies this knowledge in real-life scenarios encompassing questions requiring complex reasoning involving one-to-many relations and/or require multi-step logical intersections (explained in detailed in Section 1). To address this gap, we introduce a new benchmark, CompKE: **Comp**lex Question Answering under **K**nowledge **E**diting, encompassing 11,921 complex questions conforming to real-life scenarios. In addition, we also propose GDecom-CQA: **G**eneric **Decom**position based **C**omplex **Q**uestion **A**nswering, a novel approach tailored at complex question answering. We performed comprehensive evaluation of the GDecom-CQA using CompKE along with existing benchmarks to showcase the effectiveness of key contributions made in this work. Experimental evaluation reveal that GDecom-CQA outperforms the best-performing baseline models on CompKE by improving the Augmented-Accuracy metric by 38.5% on average.

1 Introduction

Despite large language models (LLMs) being powerful to solve a wide range of real-world scenarios, they often generate erroneous or outdated knowledge (Wang et al., 2023c; Zhang et al., 2024b).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., 2023c; Zhang et al., 2024b). A key challenge for the KE methods is their application of new knowledge to reason on multi-hop question answering (MQA)—also known as MQA under KE. MQA under KE requires multiple reasoning steps to come up with the final answer. 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, *i.e.,* *(i)* identifying who is the current president of U.S., and *(ii)* determining the spouse of the president.

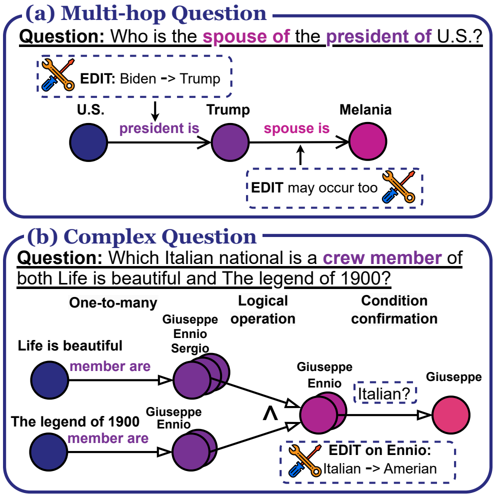


Figure 1: (a) An example of a multi-hop question involving only one-to-one sequential step-by-step reasoning. (b) An example of a complex problem involving one-to-many knowledge mapping, logical operations, and conditional confirmation.

Existing work on MQA under KE is primarily divided into two types: *(i)* memory-based methods, which place the retrieved edits before the question for in-context editing (Mitchell et al., 2022; Zhong et al., 2023); and *(ii)* parameter-based methods, which locate and modify the model parameters bases on knowledge stored in LLM (Meng et al., 2022a,b; Hu et al., 2024). Research shows memory-based methods usually outperform the parameter-based ones (Zhong et al., 2023; Gu et al., 2023; Cheng et al., 2024). These methods primarily employ a plan-and-solve paradigm (Khot et al., 2022; Wang et al., 2023a) that relies on manually defining a *task decomposition* stage to guide the LLMs to break down multi-hop questions into a series of sub-questions. For example, in Figure 1 (a), the model needs to retrieve information about the president in the first step in order to generate responses about the president's spouse.

However, existing best performing solutions for MQA under KE, *i.e.,* memory-based methods tend to be overly simplistic, often reducing the question-answering process to a mere linear chain of facts that allows step-by-step knowledge inference by mapping one entity to another through a chain of relations. These approaches are hard to adapt to real-life complex and varying situations often involving one-to-many relations. An example in this regard is shown in Figure 1 (b), which illustrates a relatively complex reasoning question: *Which Italian national is a crew member of both "Life Is Beautiful" and "The Legend of 1900?"*. Compared with classical multi-hop questions, answering this question requires: *(i) Knowledge about one-to-many relations*: A film typically features numerous participants, including directors, writers, actors *etc.,* manifesting the need to model one-to-many relation patterns in order to identify individuals associated with two movies. *(ii) Logical operations*: After having identified the individuals associated with two movies, there is a need for an appropriate intersection and/or logical operations to figure out the common participants who contributed to both films. *(iii) Condition confirmation*: After identifying the common participants between the two films, the next step is to select the individuals who meet the specified criteria, such as nationality. In the rest of the paper, we call such complex reasoning questions as the *"complex question"*.

We observe that existing memory-based methods when employed for the complex questions pose following limitations: *(i) Omission phenomenon*, *i.e.,* these methods tend to generate incomplete plans during the planning phase and often ignore different features of the complex questions required to correctly answer the question. For instance, they only adhere to one-to-one knowledge mapping and do not perform logical operations, *e.g.,* logical intersection *etc.,* to aggregate different information units. This issue arises because their prompts for decomposition only include linear multi-hop questions, and features not present in the demonstration examples are easily ignored. However, simply adding more examples to the demonstration is insufficient, as there may be many different types of complex questions. Attempts to cover all possible scenarios would result in an excessive number of prompts, leading to computational overhead and even loss of accuracy. *(ii) Incomplete retrieval, i.e.,* previous works select the top most relevant edit *w.r.t* semantical similarity when retrieving edits relevant to a sub-question. However, this approach may miss some edits, especially when the sub-question involves multiple edits. For example, the condition confirmation step requires determining whether multiple entities are Italian. When multiple people's nationalities change, multiple edits need to be retrieved, and simply selecting the top edit will result in incomplete retrieval.

To fill the gap, in this paper, we propose a new benchmark and a new memory-based editing method for complex questions. Specifically, we propose: *(a)* a new benchmark for complex questions, *i.e.,* COMPKE: **Comp**lex Question Answering under **K**nowledge **E**diting. COMPKE is curated using Wikipedia and encompasses a total of 11,921 complex questions; *(b)* a new memory-based editing approach GDECOM-CQA: **G**eneric **Decom**position based **C**omplex **Q**uestion **A**nswering, which can dynamically construct efficient decomposition prompts based on the complex questions, and can effectively solve the problem of incomplete retrieval.

We performed comprehensive evaluations to demonstrate and/or showcase the effectiveness of key contributions made in this paper. Experimental evaluation reveals that for COMPKE, GDECOM-CQA consistently outperforms other baseline models by 38.5% on average in terms of Augmented-Accuracy as evaluation metric. Moreover, on the MQUAKE-CF and MQUAKE-T dataset, GDECOM-CQA still maintains a high score, outperforming other memory-based methods by an average of 21.54% in terms of multi-hop accuracy on GPT-4O-MINI.

2 Related Work

Existing research on KE can be classified into parameter-based and memory-based methods.

Parameter-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 often perform poorly on multi-hop or complex questions and cannot be applied to closed-source models like OpenAI GPTs, which are only accessible via APIs. Additionally, these methods are more computationally expensive than the memory-based methods.

Memory-Based Methods. These 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., 2023a) 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. While these methods outperform parameter-based approaches for MQA under KE, they struggle with complex real-world scenarios. Unlike existing solutions, GDECOM-CQA augments the ability of KE for answering complex questions. It does not rely on a fixed set of decomposition prompts, instead it dynamically constructs the demonstration prompts based on the question under consideration to help solve a diverse set of complex questions.

More details about related work are provided in the Appendix A.

3 Preliminaries

Notations. 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, actor are, $\{$Worthington, Saldana,$\cdots\}$).

3.1 Complex Questions

Motivated by the example in the introduction, we formally define the complex questions that will be studied in this paper. A quick recap on multi-hop question answering (MQA) and MQA under KE is provided in Appendix B.1. We define 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 the knowledge graphs (used to map one entity s_i to another s_j via relation r), reasoning links offer extended operations by allowing conditional confirmation and logical operations, formally explained below.

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

(a) Knowledge-related Links: This type of links are used to traverse the entities 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 further divide these links into:

(i) Knowledge mapping. For S_i, we consider a knowledge mapping link as a map 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 some relation r.

(ii) Condition Confirmation. Given r and s', this link aims to identify a set of entities $S_j = \{s \in S_i \mid C(s, r, s') = True\}$ conforming to $C(s) = [(s, r, s') \in \mathcal{D}]$, which is used to examine whether s can obtain s' via r. For instance, "Whether Giuseppe and Ennio are Italian?", S_i are $\{$Giueppe, Ennio$\}$, s' is Italian and r is nationality.

(b) Logical Links: Given a set of intermediate entities $\{S_1, S_2, \cdots, S_n\} \in \mathbf{S}$, this type of reasoning link performs logical operations among the elements of S_i. For this, we use:

(i) Intersection. Intersection operation is used to determine the set of adjacent entities $S_j = \cap_{k=1}^{n} S_k$, which includes only those entities that are common across all sets.

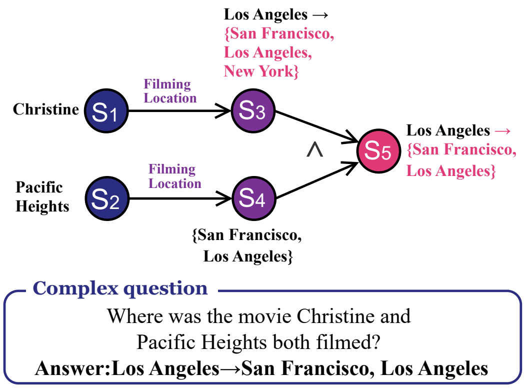

Los Angeles → {San Francisco, Los Angeles, New York}
Christine
S1
Filming Location
S3
∧
S5
Los Angeles → {San Francisco, Los Angeles}
Pacific Heights
S2
Filming Location
S4
{San Francisco, Los Angeles}
Complex question
Where was the movie Christine and Pacific Heights both filmed?
Answer:Los Angeles→San Francisco, Los Angeles

Figure 2: An example of complex question under knowledge editing. {San Franciso, Los Angels,...} represents knowledge edit.

(ii) Union. Union operation is used to compute the set of adjacent entities $S_j = \cup_{k=1}^{n} S_k$, encompassing all entities present in any of the sets.

Example. An example complex question with reasoning links is shown in Figure 2. It shows the question: *"Where was the movie Christine and Pacific Heights both 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={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 to obtain final answer, *i.e.,* $S_5 = (S_3 \cap S_4)$.

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 is denoted as $\mathcal{D}^{\mathcal{E}}_{del} = \{(s_i, r_i, \mathcal{O}_i) \mid e_i \in \mathcal{E}\}$, and the newly added knowledge is represented by $\mathcal{D}^{\mathcal{E}}_{add} = \{(s_i, r_i, \mathcal{O}'_i) \mid e_i \in \mathcal{E}\}$, The goal is to update the model's knowledge by $\mathcal{D}'$, define as: $\mathcal{D}' = (\mathcal{D} - \mathcal{D}^{\mathcal{E}}_{del}) \cup \mathcal{D}^{\mathcal{E}}_{add}$. This updated knowledge, $\mathcal{D}'$, is then used to answer the complex question Q.

4 CompKE

Although the complex questions we introduced are ubiquitous in real life, they are less studied in LLM question answering under KE. We observe that existing benchmarks primarily focus on linear multi-hop questions, making them inadequate for the comprehensive evaluation of complex questions. To bridge this gap, we propose CompKE: **Comp**lex Question Answering under **K**nowledge **E**diting. The statistics of CompKE is shown in Table 3. It comprises a total of 11,921 complex questions, each requiring up to 5 reasoning hops.

4.1 CompKE: Process-flow

We outline the workflow of our data construction process in Appendix Figure 4, with details explained as follows.

Collecting Relation Templates. We first select one-to-many relations from Wikidata, such as family-child, book-authors, movie-actors *etc.*, to be used in one-to-many knowledge mapping. Next, we collect relations that are directly associated with the characteristics of the entities, such as gender, nationality, *etc.,* to be used in conditional confirmation. In CompKE, we collected an adequate number of relations to ensure the diversity of the dataset. We provide the list of relation templates used for CompKE in Appendix Table 14.

Sampling Facts. After relation templates, we need to construct knowledge base $\mathcal{D}$. For this, we want the knowledge to be included in the questions to be relatively common rather than obscure. Based on the collected relation templates, we sample single-hop knowledge triples from Wikidata and rank them according to their frequency of access, with more frequently accessed triples placed at the top. We then use GPT-J to filter out the knowledge that the model cannot recall. We will use this collected knowledge $\mathcal{D}$ to curate complex questions.

Constructing Complex Questions. We observe that complex questions may be organized as a reasoning structures, such as the example in Figure 2, which first undergoes knowledge mapping and followed by logical operations, *e.g.,* intersection. To collect these common reasoning structures, we start by manually constructing a subset of high-quality complex questions to act as seed. Next, we remove the intermediate entities from these questions to extract the underlying reasoning structure. We use these reasoning structures a template to generate specific complex questions by instantiating it with real-world facts from $\mathcal{D}$. The process proceeds as follows: we randomly initialize the leaf nodes of the reasoning structure. From there, we use logical operation or knowledge in $\mathcal{D}$ to progressively identify the intermediate entities at next step. This process is repeated iteratively until all entities, including the intermediate ones, are fully determined.

To ensure the practical relevance of the instantiated questions, we filter out cases exhibiting the following conditions: *(i)* questions with no answer; *(ii)* questions that result in an empty set of intermediate entities; *(iii)* cases where the entities involved in the logical operations are of different types, making the logic incompatible *etc.* For illustration, we show some exemplar relational structures along with complex questions instantiated from them in the Appendix (Figure 11).

Introducing Edits. In order to simulate the knowledge edits, we construct counterfactual knowledge updates. For each complex question, we randomly select knowledge mappings and/or condition confirmations with knowledge of the form: $(s, r, \mathcal{O})$. We introduce edit $e = (s, r, \mathcal{O}')$ considering three different types of operations, *i.e., (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.

More details on constructing the dataset are provided in the Appendix C.

5 GDECOM-CQA

In this section, we introduce our proposed framework, GDECOM-CQA: **G**eneric **Decom**position based **C**omplex **Q**uestion **A**nswering, GDECOM-CQA uses a plan-and-solve paradigm, whose workflow is summarized as follows. *(i) Planning:* Construct a decomposition prompt to decompose the complex question into a series of smaller sub-questions. *(ii) Solving:* For the entities involved in each sub-question retrieve relevant edits and prompt the model to integrate both internal and external knowledge to solve each sub-question.

5.1 Planning

The planning stage of GDECOM-CQA aims to break down each complex question into multiple sub-questions to be addressed individually. However, we observe it is relatively hard to decompose complex questions compared to that of the multi-hop questions, as: *(i)* Complex questions are more general compared to the multi-hop questions exhibit greater variations. They require more decomposition examples as demonstration prompts for the LLMs. *(ii)* Unlike multi-hop questions which have a linear sequential relationship, the dependencies between sub-questions are more complex.

Decompose Complex Question. Formally, we define a decomposition of complex question Q as $Q \mapsto \{q_1, q_2, \cdots q_n\}$, as follows:

$$\{q_1, q_2, \cdots, q_n\} = \texttt{LLM}(Q, \mathcal{P}_{\text{plan}}) \tag{1}$$

where $\mathcal{P}_{\text{plan}}$ is an in-context learning prompt (with multiple example demonstrations, explained in the following subsection); q_i represents a sub-question, and $\texttt{LLM}$ is the language model used for decomposition. We represent each sub-question as $q_i = (t, i, d)$, where t represents the reasoning link type (*e.g.,* intersection, knowledge mapping), i is the specific instruction describing the detailed action or query to be executed in this sub-question, d represents the set of preceding sub-questions.

Decomposition Prompts ($\mathcal{P}_{\text{plan}}$). For $\mathcal{P}_{\text{plan}}$, we use multiple different demonstrations helpful for the effective decomposition of the complex questions. While in multi-hop questions, $\mathcal{P}_{\text{plan}}$ usually requires a small number of examples to achieve effective decomposition (Zhong et al., 2023; Gu et al., 2023), we observe for complex questions it is necessary to increase the number of examples to cover a broader and diverse range of scenarios. At the same time, including too many examples is impractical, as it leads to significant computational overhead. Moreover, an excessive number of examples may reduce the effectiveness of the decomposition, as the model's ability to understand context is inherently limited. To overcome this, instead of using a set of pre-defined examples, we use an automated way to construct a different $\mathcal{P}_{\text{plan}}$ for each Q in a way that: *(i)* we use an appropriate number of examples to avoid computational overhead and ensure effective decomposition, and *(ii)* examples in $\mathcal{P}_{\text{plan}}$ are similar to Q to the best possible extend. The process of demonstration selection for each Q is explained below.

Demonstration Selection. For demonstration selection, we first use an external memory specifically for storing decompositions. This memory is dynamically maintained, allowing new decomposition to be added in the future. For the complex question Q to be decomposed, we will select similar decomposition examples from the memory to come up with $\mathcal{P}_{\text{plan}}$. For this, we use the semantic similarity (*i.e.,* inner product of embedding vectors via obtained from embedding model), between the examples in the memory and Q, in order to select the top-k examples as $\mathcal{P}_{\text{plan}}$.

5.2 Solving

This stage of GDECOM-CQA involves retrieving relevant edits helpful for the model to solve sub-questions. While, the retrieval process for multi-hop questions is relatively straightforward, retrieving at most one edit per sub-question. In contrast, complex problems often involve a set of entities (S_i) as the intermediate answer, which may lead

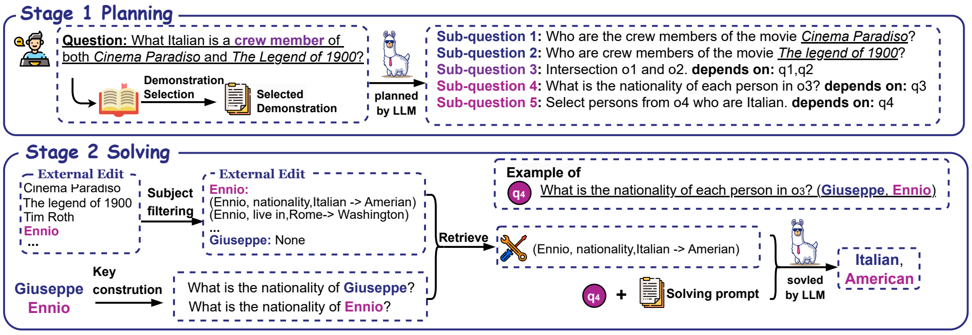

Stage 1 Planning
Question: What Italian is a crew member of both Cinema Paradiso and The Legend of 1900?
Demonstration Selection
Selected Demonstration
planned by LLM
Sub-question 1: Who are the crew members of the movie Cinema Paradiso?
Sub-question 2: Who are crew members of the movie The legend of 1900?
Sub-question 3: Intersection o1 and o2. depends on: q1,q2
Sub-question 4: What is the nationality of each person in o3? depends on: q3
Sub-question 5: Select persons from o4 who are Italian. depends on: q4
Stage 2 Solving
External Edit
Cinema Paradiso
The legend of 1900
Tim Roth
Ennio
...
Subject filtering
External Edit
Ennio:
(Ennio, nationality,Italian -> Amerian)
(Ennio, live in,Rome-> Washington)
...
Giuseppe: None
Giuseppe
Ennio
Key constrution
What is the nationality of Giuseppe?
What is the nationality of Ennio?
Retrieve
Example of
q4
What is the nationality of each person in o3? (Giuseppe, Ennio)
(Ennio, nationality,Italian -> Amerian)
q4
+
Solving prompt
sovled by LLM
Italian,
American

Figure 3: The overall workflow of GDECOM-CQA. **Planning:** Dynamically construct a decomposition prompt to decompose complex question into a series of sub-question; **Solving:** Solve each sub-question to come up with the final response.

to retrieving multiple knowledge edits from the edit memory at the same time. In this section, we first introduce the retrieval process, followed by how integrate model's own knowledge and the retrieved edits to solve the sub-question.

Suppose we want to solve q_i, in order to solve the incomplete retrieval challenge, we initially identify the set of entities that need to retrieve external information from added knowledge base $\mathcal{D}^{\mathcal{E}}_{add}$, represented by S_{q_i}, computed as $S_{q_i} = \{s \mid s \in o_j(q_j) \quad \forall q_j \in d_i\}$, where d_i represents the set of preceding sub-questions that q_i depends on, $o_j(q_j)$ represents the response output of q_j, which is a set of entities, explained in Equation 4. We retrieve external edits independently for each $s \in S_{q_i}$. This retrieval process is primarily divided into the following three steps:

Subject Filtering. In order to retrieve effectively, we start by filtering out knowledge in $\mathcal{D}^{\mathcal{E}}_{add}$ for which subject does not match s to get subset $\mathcal{D}^{s}_{sub}$.

To achieve this, previous work (Cheng et al., 2024) used aliases of entity s to see if they appear in the edit string. However, this approach may fail to retrieve all relevant edits due to incomplete alias information. To address this issue, we compute the embedding similarity between s and the subject entities in $\mathcal{D}^{\mathcal{E}}_{add}$ compared against a threshold θ to get the subset $\mathcal{D}^{s}_{sub}$, as follows:

$$\begin{aligned}\mathcal{D}^{s}_{sub} = \{(s_i, r_i, \mathcal{O}_i) \mid (s_i, r_i, \mathcal{O}_i) \in \mathcal{D}^{\mathcal{E}}_{add} \\ \wedge\ \texttt{sim}(s_i, s) \geq \theta\}\end{aligned} \tag{2}$$

where $\texttt{sim}$ is the $\texttt{cosine}$ similarity between their embedding, obtained by the encoder. We observed, this approach reduces the chances of related edits being filtered out. Note, we use all-MiniLM-L12-v2 as our embedding encoder.

Query Key Construction. The next step is to find the precise edit from $\mathcal{D}^{s}_{sub}$ by locating the edit which is most relevant to the instruction of the q_i. While, previous work use sub-questions as the key to query and compare its semantic similarity with the string form of edits to come up with most relevant edit. We observe, directly using the sub-questions for the complex question as the key will lead to noise. For instance, the sub-question: *"What is the nationality of Giuseppe and Ennio?"*, will retrieve edits like: {The author of *Giuseppe and Ennio* is Tom}, where *Giuseppe and Ennio* represents a book. We argue, in order to retrieve edits for Giuseppe, removing other entities, *e.g.,* Ennio will help improve retrieval efficiency.

For this, we construct a distinct key for each entity $s \in S_{q_i}$, represented by k_s. For k_s, we use the i, *i.e.,* the instruction part of $q_i = (t, i, d)$ by replacing its entity with the specific entity s.

Later, in order to find relevant edit for s, we use k_s as query key to iterate $\mathcal{D}^{s}_{sub}$, as follows:

$$e^{*}_{s} = \underset{(s_i, r_i, \mathcal{O}'_i) \in \mathcal{D}^{s}_{sub}}{\operatorname{argmax}} \texttt{sim}(\texttt{S}(s_i, r_i, \mathcal{O}'_i), k_s), \tag{3}$$

where $\texttt{S}(s_i, r_i, \mathcal{O}'_i)$ is the unstructured string form of the edit, e^{*}_{s} is the most relevant edit for entity s. We integrate the most relevant edits for each entity, as: $E_{q_i} = \{e^{*}_{s} \mid s \in S_{q_i}\}$.

Chain-of-thought Solving. Finally, we use the $\texttt{LLM}$ to solve the sub-questions q_i based on the retrieved edits E_{q_i}, as shown below:

$$o_i = \texttt{LLM}(\mathcal{P}_{\text{solve}}, q_i, E_{q_i}), \tag{4}$$

where $\mathcal{P}_{\text{solve}}$ is the in-context learning prompt for solving, outlined in Appendix Table 13, yielding a set of entities o_i as the response for q_i. For this, we employ a chain-of-thought reasoning method to integrate both external edits and the model's internal knowledge, as shown in *Thinking Process* in Appendix (Table 13). We find that this approach solves the problem of the model being unwilling to accept new knowledge and can effectively perform self-checking (Zhong et al., 2023) to judge whether the edit is relevant to the sub-question.

Method	CompKE					
	1-edited		100-edited		All-edited	
	Aug-Acc	Ret-Acc	Aug-Acc	Ret-Acc	Aug-Acc	Ret-Acc
Qwen2.5-3B-Instruct						
ROME	12.61	17.91	4.8	4.40	0.82	1.59
MEMIT	**20.99**	**23.86**	7.8	6.73	1.52	3.75
MeLLo	5.40	2.25	3.06	3.39	0.69	2.00
PoKeMQA	0.46	0.46	0.42	0.38	0.71	0.61
GDecom-CQA (Ours)	8.65	12.42	**7.91**	**10.19**	**7.12**	**10.40**
Qwen2.5-7B-Instruct						
ROME	22.82	25.09	7.50	7.98	0.73	0.98
MEMIT	29.40	27.72	24.11	24.80	1.88	2.05
MeLLo	17.78	13.38	10.35	17.32	8.98	12.59
PoKeMQA	3.95	3.41	2.17	1.67	2.04	1.15
GDecom-CQA (Ours)	**36.71**	**44.61**	**29.29**	**39.37**	**34.17**	**42.78**
LLaMa-3.1-8B-Instruct						
ROME	7.44	24.84	1.50	1.14	0.56	0.61
MEMIT	4.90	33.22	5.00	29.27	5.03	29.20
MeLLo	14.06	17.95	9.17	17.84	8.98	14.17
PoKeMQA	1.40	2.10	0.98	1.85	0.45	1.73
GDecom-CQA (Ours)	**39.19**	**39.09**	**31.58**	**34.86**	**31.71**	**37.41**
GPT-3.5-turbo						
MeLLo	49.21	44.88	37.10	44.09	32.61	38.58
PoKeMQA	23.20	25.15	21.47	23.28	20.20	22.20
GDecom-CQA (Ours)	**67.22**	**69.00**	**63.18**	**68.19**	**61.67**	**65.63**
GPT-4o-mini						
MeLLo	22.07	25.19	20.31	23.62	18.75	22.14
PoKeMQA	36.60	42.33	35.42	41.35	28.36	35.02
GDecom-CQA (Ours)	**64.72**	**71.20**	**62.35**	**68.92**	**60.44**	**67.53**

Table 1: Experimental results for CompKe. We **boldface** overall best results with the second-best underlined.

6 Experimentation

In this section, we extensively evaluate and compare GDecom-CQA against different baselines.

6.1 Experimental Settings

Datasets. For experimentation we use our newly proposed dataset, *i.e.,* CompKe, as well as existing publicly available datasets: MQuAKE-CF and MQuAKE-T (Zhong et al., 2023). The statistics of the datasets are provided in Appendix D.1.

Baselines. For performance comparison, we use existing best performing methods for MQA under KE as baselines. These include the memory-based variants: MeLLo (Zhong et al., 2023), and PokeMQA (Gu et al., 2023); as well as the parameter-based variants: ROME (Meng et al., 2022a), and MEMIT (Meng et al., 2022b).

Evaluation Metrics. For CompKe, we use: *(i) Augment-Accuracy (Aug-Acc):* the number of new entities used to augment the answer list after the knowledge edit that are correctly answered, compared to the original list. *(ii) Retain-Accuracy (Ret-Acc):* the number of entities that appear in both the original and edited answer lists, reflecting the model's ability to preserve unmodified knowledge. For MQuAKE we use *(iii) Multi-hop Accuracy (M-Acc)*, *i.e.,* the accuracy of the final answer of multi-hop question as the evaluation metric. Detailed mathematical formulation of these metrics are provided in Appendix D.3.

Experiment Setup. We conduct experiments under varying scales of knowledge edits, *i.e.,* using a batch of k-edits at a time with $k = \{1, 100, 1000, all\}$. For knowledge eding, we use LLaMa-3.1-8B-Instruct (Abhimanyu Dubey et al., 2024), Qwen2.5-3B-Instruct (Team, 2024), Qwen2.5-7B-Instruct (Team, 2024), GPT-3.5-turbo, and GPT-4o-mini (Achiam et al., 2023) as the target LLMs. 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 experimental setting are provided in Appendix D.

6.2 Experimental Results

The results of GDecom-CQA compared against the baseline models are shown in Tables 1 and 2. These results show that for CompKe, GDecom-CQA outperforms the baseline models by a significant margin in most cases while maintaining a comparable performance on MQuAKE.

For example, when considering the CompKe dataset and Aug-Acc as the evaluation metric, GDecom-CQA showed an average improvement of 36.60%, 70.29% and 89.11% compared to MeLLo for $\{1, 100, All\}$-edited on GPT-3.5-turbo respectively, and 76.83%, 76.03%, and 113.12% on GPT-4o-mini compared to MeLLo. We provide a detailed analysis of these experimental results in the following sections.

Smaller Models. For models with smaller parameters, such as Qwen2.5-3B-Instruct, memory-based methods perform worse than parameter-based methods. There are two main reasons for this: (i) models with smaller parameters have a limited instruction-following ability and struggles to adhere to the required format for planning; and (ii) during the solving stage, these models are unable to effectively integrate model's internal knowledge with external edits to address sub-questions. An example in this regard is the baseline model: PokeMQA, which requires a higher instruction-following capability, performs poorly on both LLaMa-3.1-8B-Instruct, and Qwen2.5-3B-Instruct. It emphasizes that for models with smaller parameters, an effective mechanism for the decomposition (not requiring a significant instruction-following ability) plays a crucial role in the end-performance of the model.

Batch Editing (#k-edits). We observe that the memory-based methods show a decline in the performance with the increase in the number of edits (k). However, for GDecom-CQA this decline is relatively lower compared with other memory-

Method	MQuAKE-CF-3K		MQuAKE-T	
	1-edited	All-edited	1-edited	All-edited
Qwen2.5-3B-Instruct				
ROME	8.5	-	29.8	-
MEMIT	9.5	1.2	33.5	1.8
MeLLo	6.3	2.8	**42.1**	**35.2**
PoKeMQA	1.9	1.5	3.1	2.2
GDecom-CQA (Ours)	**27.2**	**19.6**	40.6	30.6
Llama-3.1-8B-Instruct				
ROME	4.1	-	20.5	-
MEMIT	4.5	2.8	24.8	2.3
MeLLo	18.6	12.4	38.8	33.0
PoKeMQA	2.1	1.7	2.8	1.9
GDecom-CQA (Ours)	**38.8**	**23.6**	**65.8**	**65.0**
GPT-3.5-turbo				
MeLLo	57.4	35.3	**88.1**	**74.5**
PoKeMQA	**67.2**	**48.8**	78.2	68.1
GDecom-CQA (Ours)	59.5	41.4	81.8	73.1
GPT-4o-mini				
MeLLo	48.3	35.0	47.0	40.3
PoKeMQA	57.3	39.6	73.5	71.0
GDecom-CQA (Ours)	**60.2**	**47.4**	**82.3**	**78.0**

Table 2: Experiment results for MQuAKE-3K and MQuAKE-T with M-ACC as evaluation metric.

based methods, *e.g.,* MeLLo. This is evident in Figure 8, where we plot the performance of memory-based methods as a function of the number of edits. For instance, for CompKe with the GPT-3.5-turbo, from 1-edited to all-edited, Aug-Acc and Ret-Acc for MeLLo decreased by 33.74% and 14.06%, respectively.

For parameter-based methods, we observe deterioration in performance is higher compared with memory-based methods, especially when the number of edits increase beyond a limit, *e.g.,* k=100. Especially, Ret-Acc metric drops significantly, indicating that the model's original knowledge is compromised. Besides, we find that once the number of edits exceed a certain threshold, *i.e.,* $k \geq 600$, the model loses its ability to maintain coherent conversations and starts generating irrelevant output, shown in Appendix Table 12.

Performance on MQuAKE. GDecom-CQA achieves the highest performance for three of four target LLMs tested on MQuAKE-CF-3K. Even with the smaller Qwen2.5-3B-Instruct, it outperforms the parameter-based method, as multi-hop question decomposition is relatively simple and does not require strong instruction-following capabilities. We observe that PoKeMQA perform better on multi-hop questions than complex questions, especially for GPT-3.5-turbo. A possible justification for this is PoKeMQA is specially designed for multi-hop questions, but its generalization to other questions is poor.

Omission Phenomenon. We also analyze the performance of MeLLo using the original decomposition prompts provided with the model implementation. We observe that it leads to omission phenomenon in the decomposition phase, *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 9.

6.3 Ablation Study

We mainly conduct ablation studies on the following modules: *(i)* Demonstration selection, *(ii)* Chain-of-thought solving, *(iii)* Subject filtering, with results explained as follows:

***(i)* Demonstration Selection.** Results for the ablation studies for the demonstration selection are shown in Figure 7. These results show that removing demonstration selection in planning stage, decreases the Aug-Acc drops by 6.75% on average. These results confirm the effectiveness of our dynamic prompt constructing approach, emphasizing that providing example demonstrations similar to the complex question indeed help the model to achieve better decomposition ability.

***(ii)* Chain-of-thought Solving.** The results of GDecom-CQA without chain-of-thought reasoning approach are shown in Figure 9. These results show that without using a chain-of-thought approach to integrate internal knowledge and external edits, the performance drops by 37.45% on average. We attribute this deterioration in performance to two key factors: (a) The model is reluctant to incorporate new knowledge, and (b) The model struggles to determine whether the fact edits are relevant to the question.

***(iii)* Subject Filtering.** The results of GDecom-CQA for subject filtering are shown in Figure 10. We observe across all three LLMs, the performance of the model with subject filtering as the retrieval strategy using subject filtering exceeds than that of direct semantic retrieval. This step helps filter out a lot of irrelevant edits, thus improving the overall effectiveness of GDecom-CQA.

7 Conclusion

In this paper, we present the concept of complex questions, and introduce a new benchmark CompKe, along with a approach GDecom-CQA, for answering complex questions. Experimental evaluation shows GDecom-CQA outperforms the baseline models by a significant margin.

Limitations

This work poses following limitations:

- GDECOM-CQA incurs additional overhead in constructing the dynamic decomposition prompt, although it is minimal compared to the subsequent overhead.
- GDECOM-CQA uses an iterative approach for solving complex questions. Intermediate errors may propagate along the path and impact the final answer. For this, our current implementation lacks an effective mechanism for recovery from errors in the intermediate stages.

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. Mqa-keal: Multi-hop question answering under knowledge editing for arabic language.

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, Gang Lin, Haoyang Fei, Yuxuan Zhai, Lu Yu, Muhammad Asif Ali, Lijie Hu, and Di Wang. 2024. Multi-hop question answering under temporal knowledge editing. *ArXiv*, abs/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.

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.

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.

Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Wilke: Wise-layer knowledge editor for lifelong knowledge editing. *ArXiv*, abs/2402.10987.

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.

Tushar Khot, H. Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *ArXiv*, abs/2210.02406.

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.

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

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Annual Meeting of the Association for Computational Linguistics*.

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. 2023b. 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. 2023c. Knowledge editing for large language models: A survey. *arXiv preprint arXiv:2310.16218*.

Weixuan Wang, Barry Haddow, and Alexandra Birch. 2023d. 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.

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? *ArXiv*, abs/2305.12740.

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? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4862–4876, Singapore. Association for Computational Linguistics.

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 Knowledge Editing Benchmarks

KE is a critical area of research for LLMs, addressing the need to update knowledge to respond to dynamic real-world queries. Several benchmarks have been developed for evaluating KE methods. Early works like COUNTERFACT (Meng et al., 2022a) assess counterfactual updates, while ZsRE (Levy et al., 2017) and MzsRE (Wang et al., 2023d) 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., 2023b) propose an easy-to-use framework for LLMs that supports a variety of cutting-edge knowledge editing 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., 2023b) 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. Nevertheless, these benchmarks fall short in capturing real-world complexity, such as reasoning with one-to-many relations or combining entities via logical operations like intersection and union. To bridge this gap benchmark, we propose COMPKE, encompassing 11,921 questions involving complex reasoning structures, aimed at evaluating the performance of KE methods for complex questions.

A.2 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).

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 $e = (s, r, o \to o')$ 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: $S_1 \overset{L_1}{\to} S_2 \overset{L_2}{\to} \cdots \overset{L_{n-1}}{\to} S_n$, 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 4 shows the process by which we construct complex question. Figure 11 gives some examples of the structures in COMPKE and the corre-

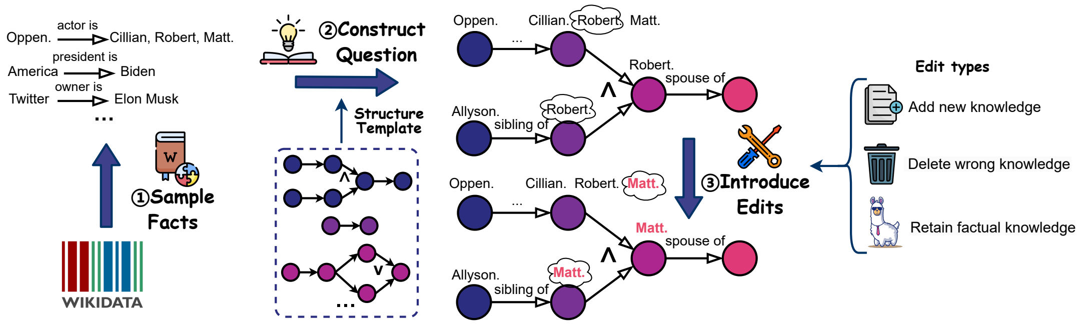


Figure 4: The construction process of COMPKE

sponding decomposition methods. Table 6 gives the SPARQL which we used to sample facts from WikiData.

D Additional Experimental Settings

D.1 Datasets

We provide the statistics of newly proposed data COMPKE in Table 3. 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.

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

Table 3: Statistical Results of COMPKE dataset.

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

$$\mathbb{E}_{q\in\mathcal{Q}}\left(\left|M'(q)\cap\mathcal{A}_{aug}\right| / |\mathcal{A}_{aug}|\right) \tag{5}$$

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-Acc) 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:

$$\mathbb{E}_{q\in\mathcal{Q}}\left(\left|M'(q)\cap\mathcal{A}_{ret}\right| / |\mathcal{A}_{ret}|\right) \tag{6}$$

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. The formula for calculating M-Acc is as follows:

$$\mathbb{1}\left[\bigvee_{q \in \mathcal{Q}} [M'(q) = a']\right]. \tag{7}$$

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 Hyper-parameters Analysis.

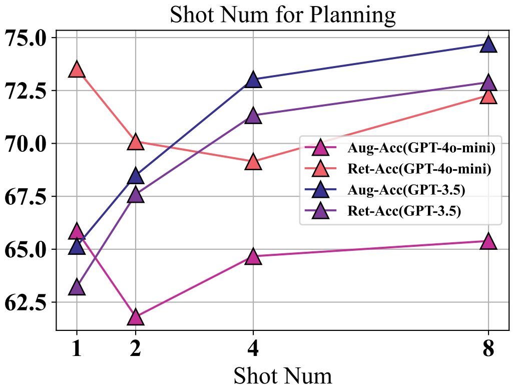


Figure 5: Hyper-parameters study on number of examples for decomposition prompt: On COMPKE with 1-edited, the curves for the ***Aug-Acc*** and ***Ret-Acc*** metrics in our method vary with the number of shots. The scores of GPT-3.5-TURBO-INSTRUCT increase monotonically as the number of shots increases, whereas GPT-4O-MINI shows insensitivity to the number of shots.

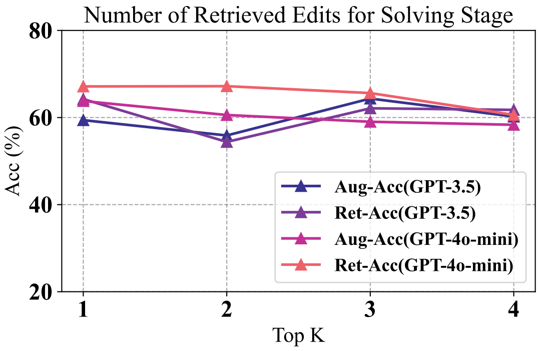


Figure 6: Hyperparameter Study of retrieved edits: In this study, conducted on COMPKE with 300 edits and the Aug-Acc metric, Top-K refers to select the top-k retrieved edits.

D.5 Experiment Setup

Table 11 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, PokeMQA, and GDECOM-CQA.

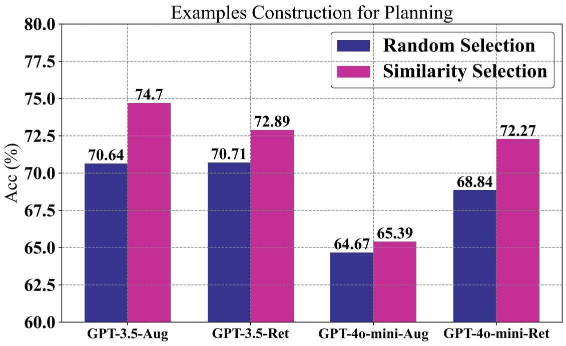


Figure 7: Ablation Study on demonstration selection for the planning stage. We compare the impact of randomly selecting examples from the demonstration memory versus selecting them based on semantic similarity.

E Additional Experimental results

E.1 Results for Batch Editing(#k-edits)

The results for the batch editing, *i.e.,* varying the number of edits (k) are presented in Figure 8.

We mainly conducted two hyper-parameters exploration experiments: the number of demonstrations retrieved during the planning phase (i.e., i-shot, where $i \in \{1, 2, 4, 8\}$) and the number of top-k edits ($k \in \{1, 2, 3, 4\}$) selected during the retrieval phase. The results are shown in Figures 5 and 6.

For GPT-3.5-TURBO-INSTRUCT we observe: both metrics, *i.e.,* Aug-Acc and Ret-Acc improve steadily as the number of demonstrations increase. However, for GPT-4O-MINI, we observe an abnormal behavior with higher scores for 1-shot and 8-shot settings while lower scores for 2-shot and 4-shot results are lower. This shows that overall GPT-3.5-TURBO-INSTRUCT shows more stable results compared to that of GPT-4O-MINI.

As the number of retrieved edits increases, overall performance decreases. However, when 3 edits are selected, performance on GPT-3.5-TURBO-INSTRUCT improves. One possible reason is that some of the matching edits fall outside the similarity range of the top-2. While using more edits can reduce the retrieval error rate, selecting too many edits may introduce additional context that interferes with the model's response. GPT-3.5-TURBO-INSTRUCT is more stable than GPT-4O-

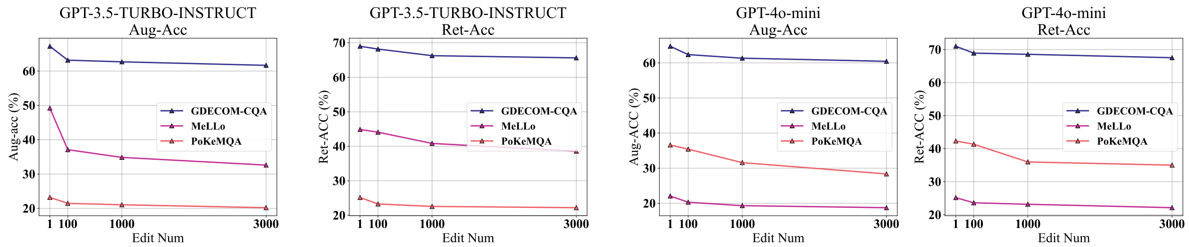


Figure 8: Performance for GDECOM-CQA, MeLLo, and PoKeMQA on the GPT-3.5-TURBO-INSTRUCT and GPT-4O-MINI, with edit number varies from 1 to 3000.

MINI in this scenario.

E.2 Ablation Study

In this section, we plot the results for the ablation experiments. We report the results for the CoT experiments in Figure 9. The results for the subject filtering experiments are reported in Figure 10. The results for the demonstration selection experiments are shown in Figure 7.

Method	1-edited		All-edited	
	Aug-Acc	Ret-Acc	Aug-Acc	Ret-Acc
GPT-3.5-TURBO-INSTRUCT				
MELLO	15.38	18.50	11.17	19.92
PokeMQA	6.18	5.11	4.72	4.44
GDECOM-CQA (Ours)	74.90	61.77	69.72	61.57
GPT-4O-MINI-2024-07-18				
MELLO	33.87	27.36	17.14	20.70
PokeMQA	10.18	22.22	9.81	17.55
GDECOM-CQA (Ours)	81.81	64.88	77.81	64.00

Table 5: Experiment Result on subset of COMPKE, which include data with incomplete retrieving challenge.

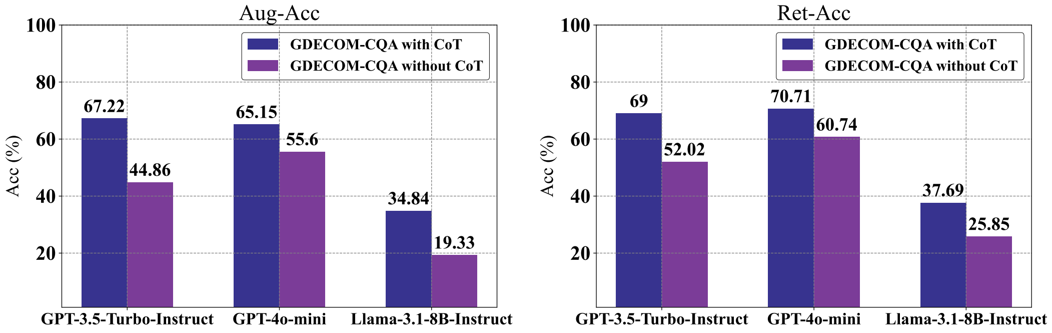


Figure 9: Ablation Study on CoT for Solving Stage: We evaluate the impact of CoT solving on COMPKE with 1-edited.

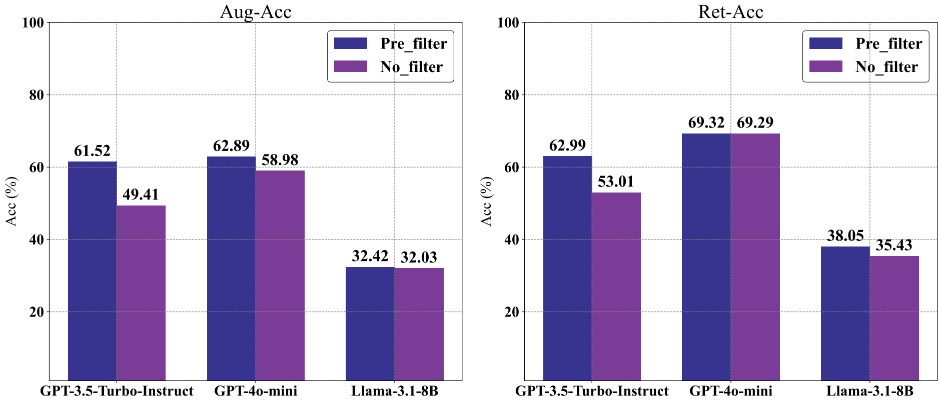


Figure 10: Ablation Study on Subject filtering strategy: This figure illustrates the impact of subject-based filtering of irrelevant edits versus direct retrieval in our method, evaluated on COMPKE with 3000-edited.

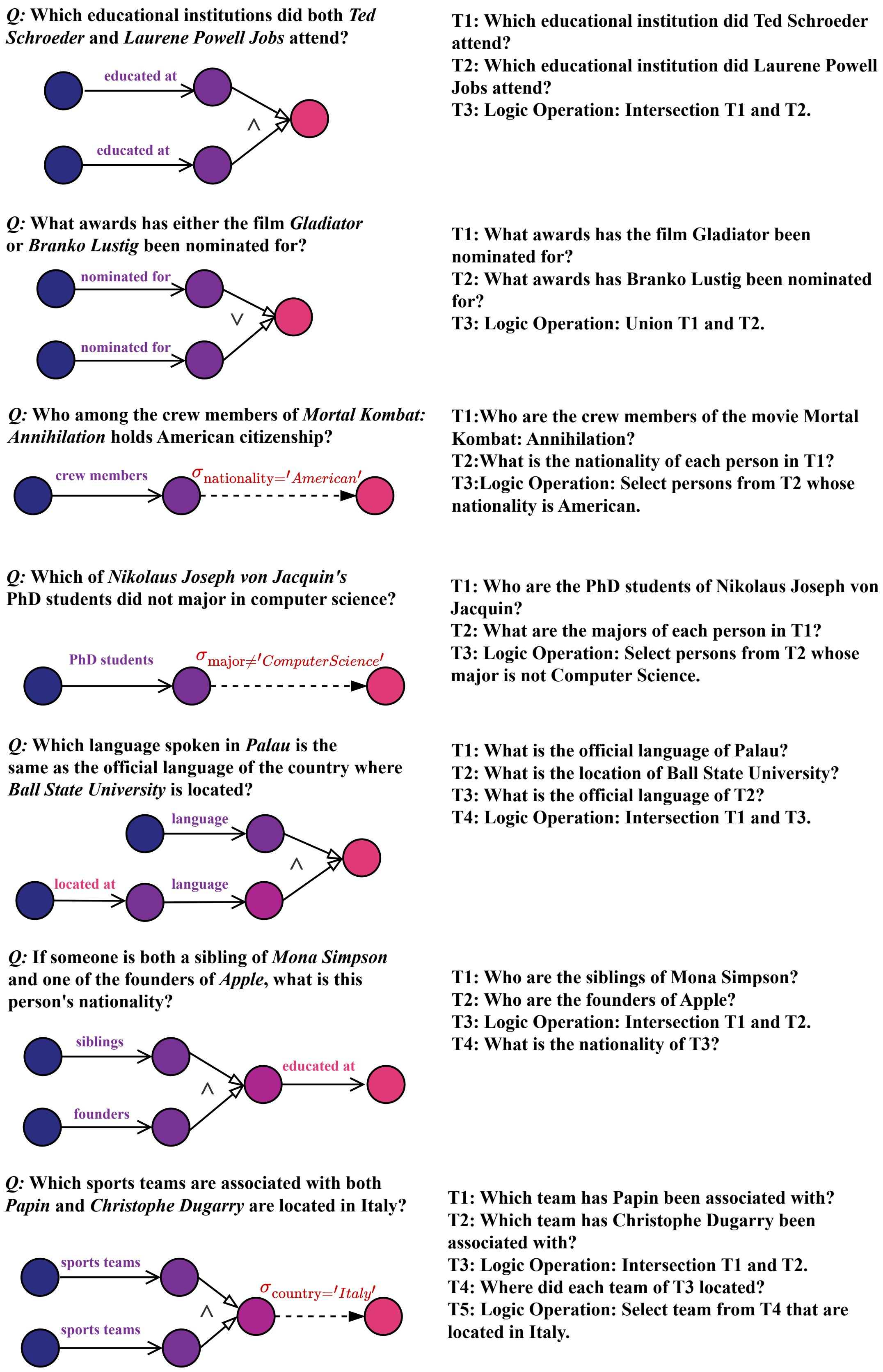
SQL Query	Description

`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 6: SPARQL Queries and Descriptions



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 11: Some typical question structure in our dataset

[8 in-context demonstrations abbreviated]

Question: **Which cast members of *Zabriskie Point* hold citizenship in the United States of America?**
PLAN STAGE:
start point: Zabriskie Point
Sub tasks:
T1: Who are the cast members of the movie Zabriskie Point? subject:Zabriskie Point
T2: What is the nationality of each person in T1? subject:T1
T3: Select persons from T2 whose nationality is USA. subject:None
Solve Stage:
Executing task T1: Who are the cast members of the movie Zabriskie Point?
context:None
Executing Retrieval:query=Who are the cast members of the movie Zabriskie Point?
Edits:[’The country to which Paul Fix belongs is France’]
Thinking Process: The question does not involve information from previous questions; the provided Updated Knowledge is relevant to the question but does not directly provide any information about the cast members of the movie ’Zabriskie Point.’ Since there is no updated information on the cast, I will use my own understanding to respond.
Answer:The cast members of the movie Zabriskie Point are: Mark Frechette, Daria Halprin, Paul Fix, Rod Taylor, Christina Hobbs, and Michael M. Murphy.
Answer List:[’Mark Frechette’, ’Daria Halprin’, ’Paul Fix’, ’Rod Taylor’, ’Christina Hobbs’, ’Michael M. Murphy’]
Executing task T2: What is the nationality of each person in T1?
context:Answer of T1:The cast members of the movie Zabriskie Point are: Mark Frechette, Daria Halprin, Paul Fix, Rod Taylor and Christina Hobbs.
Executing Retrieval:query=What is the nationality of Mark Frechette?
Executing Retrieval:query=What is the nationality of Daria Halprin?
Executing Retrieval:query=What is the nationality of Paul Fix?
Executing Retrieval:query=What is the nationality of Rod Taylor?
Executing Retrieval:query=What is the nationality of Christina Hobbs?
Edits:[’The country to which Paul Fix belongs is France’, ’The country to which Rod Taylor belongs is United States of America’]
Thinking Process: The question involves the result of T1 in the context, so I look for it in the relevant context. T1 provides the names of the cast members of the movie ’Zabriskie Point.’ The Updated Knowledge provides nationality information for Paul Fix and Rod Taylor. Since there are no updates for the other cast members, I will use my own understanding for them.
Answer: Mark Frechette is American. Daria Halprin is American. Paul Fix is French. Rod Taylor is American. Christina Hobbs is American.
Answer List: [’American’, ’American’, ’French’, ’American’, ’American’, ’American’]
Executing task T3: Select persons from T2 whose nationality is USA.
context:Answer of T1:The cast members of the movie Zabriskie Point are: Mark Frechette, Daria Halprin, Paul Fix, Rod Taylor and Christina Hobbs.
Answer of T2:Mark Frechette is American. Daria Halprin is American. Paul Fix is French. Rod Taylor is American. Christina Hobbs is American.
Edits: None
Thinking Process: The question involves filtering the information from T2 to identify individuals whose nationality is the USA. According to T2, the relevant individuals mentioned to be American are Mark Frechette, Daria Halprin, Rod Taylor and Christina Hobbs. Therefore, all individuals listed in T2 except for Paul Fix (who is French) should be included in the answer.
Answer: The persons from T2 whose nationality is USA are Mark Frechette, Daria Halprin, Rod Taylor and Christina Hobbs.
Answer List: [Mark Frechette’, ’Daria Halprin’, ’Rod Taylor’, ’Christina Hobbs’]

Table 7: An running case of GDECOM-CQA

Question: **Which cast members of *Zabriskie Point* hold citizenship in the United States of America?**

Subquestion: Who are the cast members of 'Zabriskie Point'?

Generated answer: The cast members of 'Zabriskie Point' include Mark Frechette, Daria Halprin, Paul Fix, Rod Taylor, Christina Hobbs

Retrieval: The country to which Harrison Ford belongs is France

Retrieved fact does not contradict to generated answer, so the intermediate answer is: Mark Frechette and Daria Halprin.

Subquestion: What is the country of citizenship of Mark Frechette, Daria Halprin, Paul Fix, Rod Taylor, Christina Hobbs?

Generated answer: Mark Frechette, Daria Halprin, Paul Fix, and Christina Hobbs are all from the United States, while Rod Taylor is from Australia.

Retrieval: The country to which Paul Fix belongs is France

Retrieved fact contradicts to generated answer, so the intermediate answer is: Mark Frechette, Daria Halprin and Christina Hobbs

Final answer: Mark Frechette, Daria Halprin and Christina Hobbs.

Table 8: MeLLo Failure on multi entity retrieval

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.

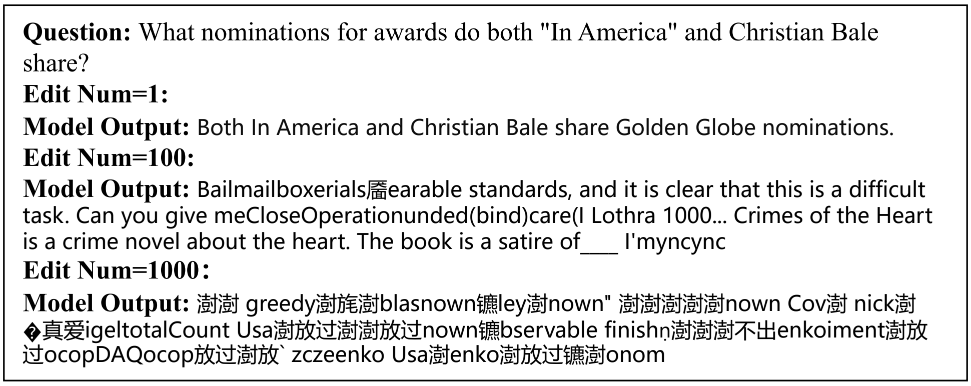
Final answer: Italy

Table 9: The decomposition of this problem by Mello did not take into account logical operations.

Question: **Which cast members of *Zabriskie Point* hold citizenship in the United States of America?**

Answer: Dennis Hopper and Brooke Shields hold citizenship in the United States of America

Table 10: MEMIT Failure on Qwen2.5-7B



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靥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�真爱igeltotalCount Usa澍放过澍澍放过nown镳bservable finishṇ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 12: 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 Qwen2.5-3B),
v_lr: 5e-1,
v_loss_layer: 31(for Llama-3.1-8B)||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 Qwen2.5-3B),
v_lr: 5e-1,
v_loss_layer: 31(for Llama-3.1-8B)||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 11: Several key hyperparameters for parameter-based KE method

```
Question: Who is someone that is both a member of the cast of the movie Birthday
Girl and a sibling of Cecile Cassel?
start point: Birthday Girl, Cecile Cassel
Sub tasks:
T1: Who are the cast members of the movie Birthday Girl? subject:Birthday Girl
T2: Who are the sibling of Cecile Cassel? subject:Cecile Cassel
T3: Logic Operation: Intersection T1 and T2. subject:None

Question: Which members of the cast of Diner are not French?
start point: Diner
Sub tasks:
T1: Who are the cast members of the movie Diner? subject:Diner
T2: What is the nationality of each person of T1? subject:T1
T3: Logic Operation: Select persons from T2 who is not French. subject:None

Question: Where were either The Abyss or Twilight Zone: The Movie filmed?
start point: The Abyss, Twilight Zone: The Movie
Sub tasks:
T1: Where was The Abyss filmed? subject:The Abyss
T2: Where was Twilight Zone: The Movie filmed? subject:Twilight Zone: The
Movie
T3: Logic Operation: Union T1 and T2 subject:None

Question: Multiple Choice: Who are the authors of the books The Farthest Shore,
Tehanu, and Tales from Earthsea?
A. Robert Harris B. Ursula K. Le Guin C. Elisabeth Hauptmann D. Anna Komnene
start point:The Farthest Shore, Tehanu, Tales from Earthsea
Sub tasks:
T1:Who is the author of The Farthest Shore? subject:The Farthest Shore
T2:Who is the author of Tehanu? subject:Tehanu
T3:Who is the author of Tales from Earthsea? subject:Tales from Earthsea
T4:Choose the correct answer(s) that have the same meaning as the authors from
T1, T2, and T3: A. Robert Harris B. Ursula K. Le Guin C. Elisabeth Hauptmann D.
Anna Komnene subject:None
```

Table 12: The prompt for **Planing Stage**, with only a portion of the prompt shown due to limited space.

Based on the latest information about Open Hearts, Valhalla Rising, Pusher II, Consider the relevant context:
None
Updated Knowledge:
The cast members of movie Open Hearts are Nikolaj Lie Kaas, Paprika Steen, Sonja Richter, Mads Mikkelsen, Jackie Chan, Michelle Yeoh, Jaden Smith, Taraji P. Henson.
please answer the following: **Who are the cast members of the movie Open Hearts?**
Thinking Process: The question does not involve information from previous questions; the provided Updated Knowledge is relevant to the question and should be taken as authoritative, so the information given should be copied directly as the answer.
Answer: The cast members of movie Open Hearts are Nikolaj Lie Kaas, Paprika Steen, Sonja Richter, Mads Mikkelsen, Jackie Chan, Michelle Yeoh, Jaden Smith, Taraji P. Henson.
Answer List: 'Nikolaj Lie Kaas', 'Paprika Steen', 'Sonja Richter', 'Mads Mikkelsen', 'Jackie Chan', 'Michelle Yeoh', 'Jaden Smith', 'Taraji P. Henson'

Based on the latest information about Courage Under Fire, CEO of Dell Technologies, Consider the relevant context:
None
Updated Knowledge:
None
please answer the following question: **Who are the cast members of the movie The Public Eye?**
Thinking Process: The question does not involve information from previous questions;The updated knowledge does not provide any information related to the issue, indicating that the information has not been updated; therefore, I will directly use my own understanding to response.
Answer: The cast member of 'The Public Eye' includes:Joe Pesci, Barbara Hershey, Jerry Adler, Richard T. Jones, William Schallert, Anne Meara, Peter Riegert, Philip Bosco.
Answer List:'Joe Pesci', 'Barbara Hershey', 'Jerry Adler', 'Richard T. Jones', 'William Schallert', 'Anne Meara', 'Peter Riegert', 'Philip Bosco'

Based on the latest information about Courage Under Fire, CEO of Dell Technologies, Consider the relevant context:
T1:The movie 'Courage Under Fire' was filmed in Austin, El Paso, Fort Hood, San Antonio, and Southern California.
T2:The CEO of Dell Technologies is Michael S. Dell.
T3:Michael S. Dell reside in Austin, El Paso.
Updated Knowledge:
None
please answer the following: **Intersection the object of T1 AND object of T3.**
Thinking Process: The question involves the result of T1 and T3 in the context, so I look for it in the relevant context.Updated Knowledge is empty, which indicates that it does not involve updated knowledge. I will directly perform logical operations to take the intersection of the answers to the two previous questions.
Answer: The Intersection of the object of T1 AND object of T3 is Austin and El Paso.
Answer List:'Austin','El Paso'

Table 13: The prompt for **Solving Stage**, with only a portion of the prompt shown due to limited space.

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 14: Relations we use to construct our dataset