# DeepEdit: Knowledge Editing as Decoding with Constraints

**Yiwei Wang**[†] **Muhao Chen**[‡] **Nanyun Peng**[†] **Kai-Wei Chang**[†]
[†] University of California, Los Angeles [‡] University of California, Davis
wangyw.evan@gmail.com
https://wangywust.github.io/deepedit.io/

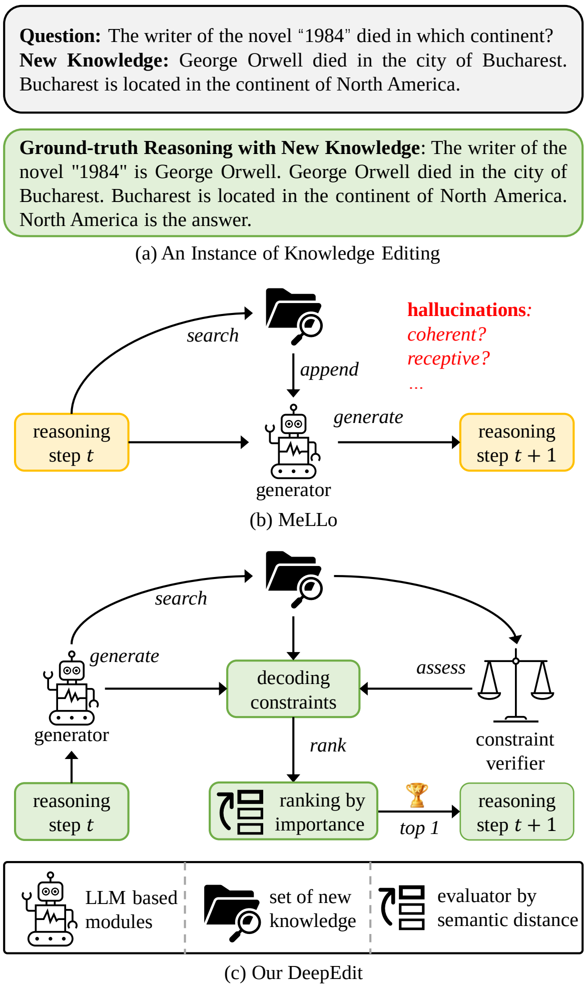
## Abstract

How to edit the knowledge in multi-step reasoning has become the major challenge in the knowledge editing (KE) of large language models (LLMs). The difficulty arises because the hallucinations of LLMs during multi-step reasoning often lead to incorrect use of new knowledge and incorrect answers. To address this issue, we design decoding constraints to "regulate" LLMs' reasoning, enhancing logical coherence when incorporating new knowledge. We propose a new KE framework: **DEEPEDIT** (*Depth-first Search-based Constrained Decoding for Knowledge Editing*), which enhances LLMs's ability to generate coherent reasoning chains with new knowledge through depth-first search. Our search selects the most important knowledge that satisfies our constraints as the reasoning step to efficiently increase the reasoning depth. In addition to DEEPEDIT, we propose two new KE benchmarks: **MQUAKE-2002** and **MQUAKE-HARD**, which provide more precise and challenging assessments of KE approaches. Qualitatively, DEEPEDIT enables LLMs to produce succinct and coherent reasoning chains involving new knowledge. Quantitatively, it yields significant improvements on multiple KE benchmarks.

## 1 Introduction

Knowledge editing (KE) aims to enable large language models (LLMs) to incorporate new or updated knowledge, in contrast to relying solely on outdated parametric knowledge. Multi-hop question answering is the challenging task to evaluate KE (Zhong et al., 2023), which requires models to answer questions corresponding to a chain of facts including new knowledge (see Figure 1(a)).

Recent literature has mainly focused on performing KE through in-context editing (Zhong et al., 2023; Cohen et al., 2023; Zheng et al., 2023), which puts new knowledge in the input context to fulfill KE (see Figure 1(b)). What new knowledge to use and how to utilize new knowledge is decided by the LLM generators' implicit reasoning. However, the LLM generator's hallucinations against the in-context new knowledge can harm the logical coherence of multi-step reasoning and result in incorrect answers (Zhang et al., 2023).



**Question:** The writer of the novel "1984" died in which continent? **New Knowledge:** George Orwell died in the city of Bucharest. Bucharest is located in the continent of North America.

**Ground-truth Reasoning with New Knowledge**: The writer of the novel "1984" is George Orwell. George Orwell died in the city of Bucharest. Bucharest is located in the continent of North America. North America is the answer.

(a) An Instance of Knowledge Editing

(b) MeLLo

(c) Our DeepEdit

Figure 1: (a) An instance of multi-hop question answering with new knowledge. (b) Prior KE methods, e.g., MeLLo (Zhong et al., 2023), suffers from the hallucinations of LLM generators. (c) Our DEEPEDIT controls LLMs with decoding constraints to produce coherent reasoning chains involving new knowledge.

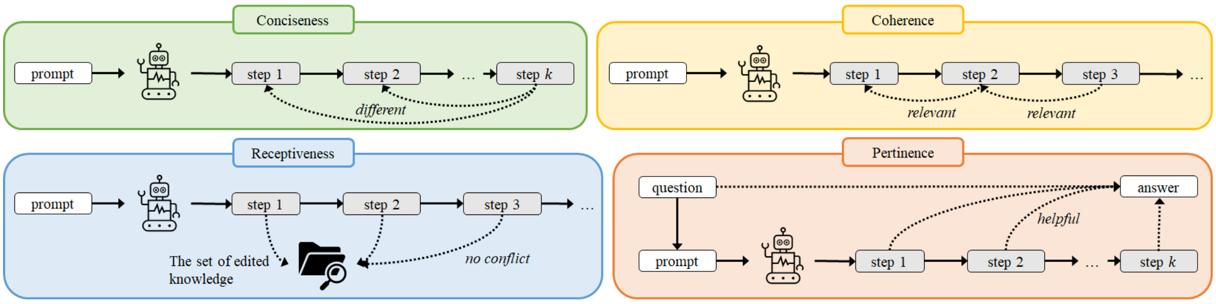


Figure 2: Our DeepEdit controls LLMs' reasoning to follow the constraints of Conciseness, Coherence, Receptiveness, and Pertinence so as to soundly incorporate new knowledge into LLMs' question answering.

To get the correct answers to multi-hop questions with new knowledge, we should ensure that the adjacent reasoning steps are logically coherent, and new knowledge is properly integrated into reasoning (see Fig. Figure 1(a). In this work, we explore using decoding constraints and search algorithms to explicitly control LLMs' reasoning with new knowledge.

The central idea of this paper is to view KE as a problem of constrained decoding. We design the decoding constraints that facilitate LLMs to soundly incorporate new knowledge in LLMs reasoning. Specifically, we propose the following constraints: Conciseness requires every reasoning step to be identical; Coherence requires the adjacent reasoning steps to be coherent; Receptiveness requires new knowledge to replace the conflicted parametric knowledge; Pertinence requires every reasoning step to be relevant to the target question (see Figure 2). A multi-step reasoning chain that leads to the correct answer should satisfy the above constraints.

We incorporate the above constraints into a new KE framework: **DeepEdit** (*Depth-first Search-based Constrained Decoding for Knowledge Editing*). When producing every reasoning step, DeepEdit searches through the parametric and new knowledge to find the knowledge items that satisfy our constraints, termed as valid step candidates. Among the valid candidates, DeepEdit prioritizes more important ones as the new reasoning step for further reasoning. We comprehensively evaluate the importance of different candidates based on their semantic relationships and the prior that the new knowledge is more informative to LLMs than the parametric knowledge.

Note that the classical depth-first search has a backtracking mechanism that enables us to traverse all the reasoning chains of valid reasoning steps (Tarjan, 1972). However, such backtracking is undoubtedly time-consuming and unnecessary after an answer is found. Therefore, we design an early-stopping mechanism to stop the depth-first search when an answer is found by LLMs. Such an early stopping significantly improves the reasoning efficiency by reducing redundant iterations, as validated by the qualitative analysis (see Section 3.3) and empirical results (see Section 5.4).

The advantages of DeepEdit are two-fold. First, DeepEdit is flexibly applicable to any black-box LLM without requiring access to model parameters or token-wise distributions. Second, DeepEdit improves both the receptiveness of new knowledge and the coherence of multi-step reasoning by directly controlling the LLMs' reasoning with decoding constraints. Unlike previous KE methods that edit either the model parameters (Zhu et al., 2020) or the input prompts (Cohen et al., 2023), we directly control the decoding of LLMs to incorporate the new knowledge into outputs, which opens up a new direction for KE in LLMs.

To provide precise and challenging assessments of KE methods, we build two new benchmarks, **MQuAKE-2002**, and **MQuAKE-hard**, which resolve the knowledge-conflicting annotation mistakes in the popular KE benchmark MQuAKE-3k (Zhong et al., 2023). We evaluate DeepEdit on MQuAKE-3k, MQuAKE-2002, and MQuAKE-hard (Zhong et al., 2023; Meng et al., 2022a). Qualitatively, DeepEdit enhances LLMs to produce succinct reasoning with new knowledge, as shown in Figure 9. Quantitatively, DeepEdit improves the KE for popular LLMs by a significant margin (OpenAI, 2022; Touvron et al., 2023).

## 2 Related Work

**Knowledge Editing** Previous studies have explored multiple methods for editing the knowledge of LLMs by introducing new knowledge into static model artifacts (Zhu et al., 2020; Sotoudeh and

Thakur, 2019; Dai et al., 2021; Hase et al., 2021; Zhou et al., 2023; Dong et al., 2022; Huang et al., 2023). Some of these approaches involve identifying and modifying model weights associated with specific concepts (Meng et al., 2022a,b; Dai et al., 2021), as well as rapid adaptation facilitated by a compact auxiliary editing network (Mitchell et al., 2022; De Cao et al., 2021; Mitchell et al., 2021). Given the fast-growing parameter sizes of large language models (LLMs) (Zhao et al., 2023), frequently updating LLMs with new knowledge through retraining is more and more expensive. Hence, it is vital to effectively edit the LLMs' knowledge without retraining. Recent work observes the superior KE performance of in-context editing. MeLLo (Zhong et al., 2023) designs a single prompt to alternately conduct the text generation and new knowledge analysis using the LLM generator. Cohen et al. (2023) propose to append the new knowledge the head of the input prompt. In-context editing lacks direct control of the LLMs' outputs and how the new knowledge influences the outputs is not traceable.

Different from the prior work, we propose a new KE method DeepEdit, which effectively augments LLMs on KE with constrained decoding. We illustrate a comparison between DeepEdit and existing KE methods in Table 1. Our DeepEdit outperforms the prior KE methods by editing the LLMs' outputs directly, which paves the way to a new direction of LLMs' KE.

**Constrained Decoding** Constrained decoding (Qin et al., 2020, 2022; Lu et al., 2020, 2021; Kumar et al., 2022; Liu et al., 2023; Geng et al., 2023) is a classical natural language processing topic, which is mainly applied to control the lexicon, sentiment or style of the generated text. FUDGE (Yang and Klein, 2021) employs weighted decoding (WD) by directly adjusting the probabilities of the vocabulary distribution with an auxiliary classifier probability. NADO (Meng et al., 2022c) suggests sampling from a vocabulary distribution similar to an approximation of the sequence-level oracle. GeDi (Krause et al., 2020) and DExperts (Liu et al., 2021) also adopt the weighted decoding approach through the training of generative classifiers. Constrained decoding methods generally require access to the LLMs' token-wise distribution. Different from them, our DeepEdit avoids the requirements of distribution access by decoding at the reasoning step level. We extend the applications of constrained decoding to KE. We design the semantic constraints to soundly incorporate new knowledge in LLMs' reasoning, and verify the complex semantic constraints with LLMs.

| KE Method | Editing<br>Model Parameters | Editing<br>Input Prompts | Editing<br>Outputs |
|---|---|---|---|
| MEND (Mitchell et al., 2021) | ✓ | | |
| MEMIT (Meng et al., 2022c) | ✓ | | |
| ICE (Cohen et al., 2023) | | ✓ | |
| MeLLo (Zhong et al., 2023) | | ✓ | |
| DeepEdit (Ours) | | | ✓ |

Table 1: Comparison between KE methods.

## 3 Methodology

Multi-hop question answering with new knowledge assumes a database that stores all new facts, which LLMs should adhere to, and posits that only a few facts can influence the ground-truth answer to each question. In this section, we will introduce how we integrate the decoding constraints into the design of a novel KE method, DeepEdit (Depth-first Search based Decoding for Knowledge Editing), to enhance the KE of black-box LLMs.

### 3.1 Constraint Designing for Knowledge Editing

On an LLM's multi-step reasoning chain to answer a multi-hop question involving new knowledge, we find the following constraints on reasoning steps can help LLMs to give the correct answer:

1. **Conciseness**: Every step is identical.
2. **Coherence**: Adjacent steps are coherent.
3. **Receptiveness**: No conflicts exist between reasoning steps and new knowledge.
4. **Pertinence**: Every reasoning step should be relevant to the target question.

Among the above four constraints, Conciseness removes loops in the reasoning chain; Coherence guarantees the coherence of reasoning; Receptiveness ensures the LLMs' receptiveness to new knowledge; Pertinence guides LLMs to answer the target question. We visualize different constraints in Figure 2. The above constraints are easy to follow for LLMs when there is no new knowledge. However, the utilization of in-context new knowledge is influenced by the LLMs' hallucinations, which can easily break the aforementioned constraints. In this work, we apply the above

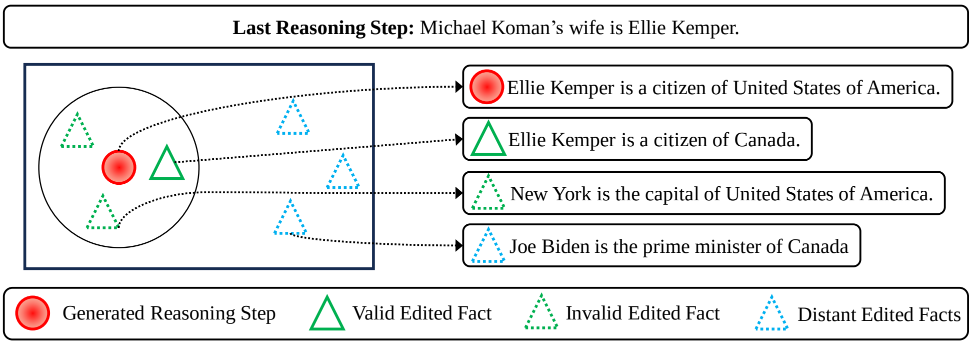

Last Reasoning Step: Michael Koman's wife is Ellie Kemper.
Ellie Kemper is a citizen of United States of America.
Ellie Kemper is a citizen of Canada.
New York is the capital of United States of America.
Joe Biden is the prime minister of Canada
Generated Reasoning Step
Valid Edited Fact
Invalid Edited Fact
Distant Edited Facts

Figure 3: When producing every reasoning step, we retrieve the edited facts that are close to the generated step in the semantic space (*left*). Only the edited facts that satisfy all the constraints, i.e., valid edited facts, are considered as the step candidates (*right*).

constraints to incorporate the new knowledge into LLMs' reasoning.

### 3.2 Constraint Verification on Every Step

We term the reasoning steps that satisfy all the constraints as valid steps. Because an invalid reasoning step would mislead LLMs into generating incorrect subsequent steps, we verify the constraints at the decoding of each reasoning step.

At every iteration, we take both the parametric and new knowledge as the step candidates. First, we let LLMs generate one reasoning step with temperature as $0$, which represents the parametric knowledge that the LLM believes to be most useful to answer the target question. Second, we retrieve $N$ new knowledge closest to the generated reasoning step. The distance between knowledge and the generated step is measured by the semantic distance of sentence-level embeddings given by a pre-trained BERT model (Devlin et al., 2019). The closer the context of the new knowledge is to the generated step, the more likely it is that the new knowledge will be helpful in answering the target question. We take these $N + 1$ knowledge as the reasoning step candidates, with examples shown in Figure 3.

We verify the constraints on the aforementioned step candidates to identify the valid ones. In principle, every constraint's verification can be seen as a binary classification problem on a pair of sentences. For example, COHERENCE's verification can be completed by assessing whether a pair of reasoning steps is relevant to each other or not. Therefore, one option to do constraint verification is to train a binary classifier for every constraint. However, such a solution is not data or time-efficient. In this work, we develop four specialized in-context learning-based LLM agents to act as the verifiers. Every verifier's prompt includes $D$ positive and negative demonstrations randomly sampled from MQuAKE (Zhong et al., 2023). In our experiments, we set $D = 2$ and $N = 2$ by default.

### 3.3 Efficient Reasoning with Depth-first Search

At every iteration, there may exist more than valid step candidates. We can consider each valid step candidate simultaneously as the new reasoning step and make further reasoning based on each of them separately. This design can be seen as a breadth-first search of a valid multi-hop reasoning chain. It is undoubtedly time-consuming because the time complexity grows exponentially with the number of reasoning steps. This raises a natural question:

*Can we increase the depth of reasoning chains more efficiently than the breadth-first search?*

Our answer is yes. We propose prioritizing more important step candidates as the reasoning step for further reasoning. Actually, different valid step candidates are not equally important. First, the edited facts should be more important than the generated step, since the former generally contains new knowledge unknown to LLMs. Secondly, the edited facts that are closer to the generated step are more important. They are more helpful in answering the target question due to their higher relevance to the context generated by the LLM's reasoning. Based on the above prior, in every iteration, we select the most important knowledge as the next reasoning step and do further reasoning based on it, as shown in Figure 4.

We name our method as DEEPEDIT: depth-first search-based constrained decoding for knowledge editing. If we follow the classical depth-first search

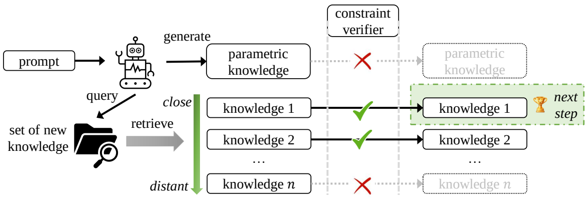


Figure 4: At every iteration, we verify the decoding constraints over parametric and new knowledge to find the valid step candidates. Then we take the most important candidate as the next reasoning step to efficiently increase the reasoning depth.

algorithm (Tarjan, 1972), we should include a backtracking mechanism to traverse all the reasoning chains of valid steps. However, such backtracking is undoubtedly time-consuming and unnecessary after an answer is found. Therefore, we design an early-stopping mechanism to stop the search when an answer is found by LLMs. We visualize DeepEdit in Figure 1 the example iteration-wise outputs in Figure 5.

Assume the number of reasoning hops as $h$, and the number of valid step candidates per iteration as $c$. The time complexity of breadth-first search is $\mathcal{O}(c^h)$. In contrast, thanks to our depth-first search design with the early-stopping mechanism, our model has a significantly lower time complexity of $\mathcal{O}(h)$ in the best case. Empirical results validate DeepEdit's superior efficiency compared with the breadth-first search (see Section 5.4).

The strategy of depth-first search has been utilized by some research on LLMs, such as tree-of-thoughts (Yao et al., 2023). Different from them, our DeepEdit's depth-first search is designed to efficiently find the valid multi-hop reasoning chain incorporating new knowledge. Our DeepEdit is free to be applied to all black-box LLMs: it bypasses the requirement of accessing the token-wise distributions because our decoding strategy is defined at the reasoning step level.

# 4 New Benchmarks for More Precise and Challenging Evaluation of KE

We provide two new benchmarks for the evaluation of KE methods: MQuAKE-2002 and MQuAKE-hard. Both are built based on MQuAKE (Zhong et al., 2023), a recent KE dataset that is designed to evaluate the KE methods on helping LLMs to answer multi-hop questions given new knowledge. Every instance in MQuAKE includes a multi-hop question and several edited facts, every of which can alter the ground-truth answer to the question.

## 4.1 Annotation Mistakes in One-Third Instances of MQuAKE

Zhong et al. (2023) suggests using a randomly sampled 3,000 instance subset of MQuAKE to do the evaluation, which reduces the experimental costs, known as MQuAKE-3k. There are two issues of using MQuAKE-3k to evaluate KE. The first issue is that the new knowledge from different instances MQuAKE-3k can cause conflicts and mistakes in the ground-truth answers. In other words, the ground-truth answer from instance $A$ can be altered by the new knowledge from another instance $B$. We show an example of knowledge conflicts in MQuAKE-3k in Figure 6. These knowledge conflicts will make the ground-truth answers incorrect given the conflicted new knowledge because the inference on every instance could retrieve the new knowledge from all instances. We observe that 998 instances' ground-truth answers are broken by the new knowledge conflicts across instances.

## 4.2 New Benchmark MQuAKE-2002 for More Precise Evaluation

To address the issue of annotation mistakes in MQuAKE-3k, we provide a new KE benchmark based on MQuAKE, which does not have any knowledge conflict across instances. This benchmark includes 2,002 instances, so we term it as MQuAKE-2002. We filter out the instances of which the ground-truth answers are broken by the new knowledge from other instances to produce MQuAKE-2002. Compared with MQuAKE-3k, our MQuAKE-2002 provides a more precise evaluation for KE methods, since it removes the annotation mistakes due to knowledge conflicts across instances. The data statistics of MQuAKE-2002 is provided in Table 2.

## 4.3 New Benchmark MQuAKE-hard for Challenging Evaluation

The second issue of MQuAKE-3k is that more than $60\%$ instances in it only contain at most two edited facts that influence the answers, which are not challenging enough to evaluate the KE methods on handling multiple edited facts that can alter the ground-truth answers. We construct a more challenging subset of MQuAKE by selecting the instances that contain the highest number of edited facts per instance excluding MQuAKE-3k. We term this challenging set as MQuAKE-hard, which includes 429 instances and every

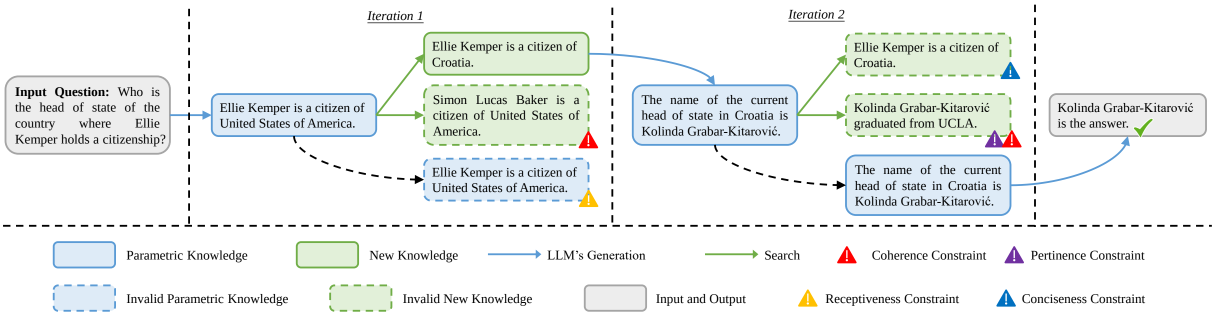
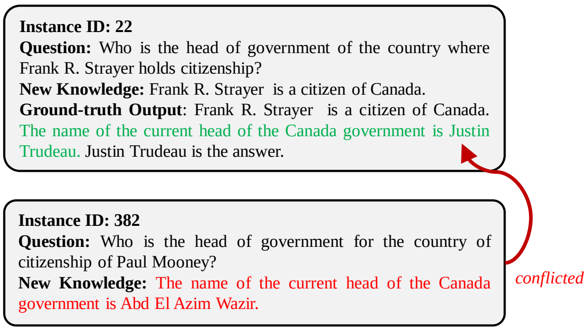

Iteration 1
Iteration 2
Input Question: Who is the head of state of the country where Ellie Kemper holds a citizenship?
Ellie Kemper is a citizen of United States of America.
Ellie Kemper is a citizen of Croatia.
Simon Lucas Baker is a citizen of United States of America.
Ellie Kemper is a citizen of United States of America.
The name of the current head of state in Croatia is Kolinda Grabar-Kitarović.
Ellie Kemper is a citizen of Croatia.
Kolinda Grabar-Kitarović graduated from UCLA.
The name of the current head of state in Croatia is Kolinda Grabar-Kitarović.
Kolinda Grabar-Kitarović is the answer.
Parametric Knowledge
New Knowledge
LLM's Generation
Search
Coherence Constraint
Pertinence Constraint
Invalid Parametric Knowledge
Invalid New Knowledge
Input and Output
Receptiveness Constraint
Conciseness Constraint

Figure 5: The iteration-wise visualization of our DEEPEDIT on an instance in MQUAKE-3K. In every iteration, the retrieved new knowledge (highlighted in green) is ranked from top to bottom as step candidates based on their semantic distances to the generated step, in ascending order.



**Instance ID: 22**
**Question:** Who is the head of government of the country where Frank R. Strayer holds citizenship?
**New Knowledge:** Frank R. Strayer is a citizen of Canada.
**Ground-truth Output**: Frank R. Strayer is a citizen of Canada. The name of the current head of the Canada government is Justin Trudeau. Justin Trudeau is the answer.

**Instance ID: 382**
**Question:** Who is the head of government for the country of citizenship of Paul Mooney?
**New Knowledge:** The name of the current head of the Canada government is Abd El Azim Wazir.

Figure 6: An example of knowledge conflicts across instances in MQUAKE-3K. New knowledge (in red) in instance 382 is conflicted with the ground-truth reasoning step (in green) of instance 22.

instance contains four edited facts that can change the ground-truth answer. The data statistics of MQUAKE-HARD is also provided in Table 2.

## 5 Experiments

In this section, we evaluate the KE performance of our DEEPEDIT method when applied to LLMs, and compare against strong KE baselines (Zhong et al., 2023). Our experimental settings closely follow those of the previous work (Zhong et al., 2023) to ensure a fair comparison.

### 5.1 Experimental Settings

We take the following KE benchmarks MQUAKE-3K (Zhong et al., 2023), MQUAKE-2002, and MQUAKE-HARD for evaluation. We follow the existing work (Zhong et al., 2023) to use accuracy as the main evaluation metric. For every instance, we test the accuracy of the first question in the instance, since different questions in the same instance are semantically equivalent and correspond to the same ground-truth answer. We follow (Zhong et al., 2023) to set the KE batch size as 100 and full batch for evaluation. The KE batch size means the number of instances that provide new knowledge for retrieval when answering every question.

**Compared methods.** We take the strong KE methods into comparison: MEND (Mitchell et al., 2021), MEMIT (Meng et al., 2022b), IKE (Zheng et al., 2023), ICE (Cohen et al., 2023), and MeLLo (Zhong et al., 2023). MEND produces weight updates by transforming the raw fine-tuning gradients given an edited fact. MEMIT updates feedforward networks in a range of layers to encode the edited facts. IKE and ICE conduct KE by adding new knowledge to the input prompts. MeLLo is the state-of-the-art KE method that prompts the LLMs to generate subquestions and append new knowledge to the prompts. The former two KE methods are only applicable to white-box LLMs, while the latter three can fit the block-box LLMs.

**Model configuration.** We set the hyper-parameters of the baseline methods as suggested by their papers. We apply the KE methods to the popular LLMs GPT-3.5-TURBO-INSTRUCT, TEXT-DAVINCI-003, and LLAMA2-7B-CHAT (OpenAI, 2022; Touvron et al., 2023) for evaluation. The former two are black-box LLMs and the latter one is a white-box LLM. By default, we set the decoding temperature as 0.0 to minimize the randomness of LLMs' outputs. Our constraint verifiers use the same LLM as the generator.

### 5.2 Overall Performance

We incorporate DEEPEDIT with several choices of LLMs to test its effectiveness on editing LLMs' knowledge. We report the test accuracy on MQUAKE, MQUAKE-3K, MQUAKE-2002 in Table 3. When the KE batch size is 100, our DEEPEDIT improves the accuracy of question answering with KE by 69%, 58%,

| Benchmark | # Instances | # Hops per Instance | # Edited Facts per Instance | # Conflicted Instances |
|---|---|---|---|---|
| MQuAKE-3k (Zhong et al., 2023) | 3,000 | 3.0 | 2.0 | 998 |
| MQuAKE-2002 (Ours) | 2,002 | 2.7 | 2.2 | 0 |
| MQuAKE-hard (Ours) | 429 | 4.0 | 4.0 | 0 |

Table 2: Data statistics of different benchmarks. # Conflicted Instances represent the number of instances of which the ground-truth labels are affected by the new knowledge from other instances. An example is shown in Figure 6.

| Method | MQuAKE-3k | | MQuAKE-hard | | MQuAKE-2002 | |
|---|---|---|---|---|---|---|
| | 100 edited | all edited | 100 edited | all edited | 100 edited | all edited |
| LLaMA2-7B-Chat w/ MEND (Mitchell et al., 2021) | 7.2 | 3.9 | 0.5 | 0.2 | 7.7 | 4.1 |
| LLaMA2-7B-Chat w/ MEMIT (Meng et al., 2022b) | 7.9 | 4.2 | 0.9 | 0.5 | 8.3 | 4.8 |
| LLaMA2-7B-Chat w/ IKE (Zheng et al., 2023) | 12.5 | 6.2 | 1.2 | 0.5 | 12.9 | 6.5 |
| LLaMA2-7B-Chat w/ ICE (Cohen et al., 2023) | 12.9 | 6.3 | 1.4 | 0.7 | 13.2 | 6.5 |
| LLaMA2-7B-Chat w/ MeLLo (Zhong et al., 2023) | 16.3 | 10.8 | 1.9 | 1.6 | 17.1 | 11.8 |
| LLaMA2-7B-Chat w/ DeepEdit (Ours) | **27.6** | **11.2** | **14.5** | **7.0** | **29.5** | **12.9** |
| GPT-3.5-Turbo-Instruct ♠ w/ IKE (Zheng et al., 2023) | 15.5 | 8.3 | 1.6 | 1.2 | 17.8 | 11.0 |
| GPT-3.5-Turbo-Instruct ♠ w/ ICE (Cohen et al., 2023) | 16.1 | 8.5 | 2.1 | 1.4 | 18.1 | 11.5 |
| GPT-3.5-Turbo-Instruct ♠ w/ MeLLo (Zhong et al., 2023) | 29.9 | 20.0 | 3.7 | 1.6 | 31.2 | 25.1 |
| GPT-3.5-Turbo-Instruct ♠ w/ DeepEdit (Ours) | **47.2** | **38.0** | **51.3** | **48.0** | **60.1** | **53.7** |
| Text-Davinci-003 ♠ w/ IKE (Zheng et al., 2023) | 18.2 | 8.9 | 1.9 | 1.2 | 19.1 | 10.2 |
| Text-Davinci-003 ♠ w/ ICE (Cohen et al., 2023) | 19.0 | 8.6 | 2.3 | 1.2 | 19.5 | 10.8 |
| Text-Davinci-003 ♠ w/ MeLLo (Zhong et al., 2023) | 32.2 | 25.6 | 4.7 | 2.3 | 33.5 | 27.8 |
| Text-Davinci-003 ♠ w/ DeepEdit (Ours) | **49.1** | **38.8** | **52.0** | **49.2** | **60.9** | **54.8** |

Table 3: Experimental results (accuracy; %) on the dataset MQuAKE-3k and MQuAKE-hard. "$k$ edited" denotes the KE batch size, which means the number of instances providing new knowledge on answering every question. Models with ♠ are closed source. The best KE result on every LLM is highlighted in **bold** font.

| LLM as Constraint Verfier | CON | COH | REC | PER |
|---|---|---|---|---|
| LLaMA2-7B-Chat | 82.3 | 77.9 | 86.3 | 78.2 |
| GPT-3.5-Turbo-Instruct | 92.8 | 91.5 | 92.2 | 86.7 |

Table 4: Experimental results (verification accuracy; %) of our verifiers for different constraints.

| GPT-3.5-Turbo-Instruct w/ DeepEdit | 49.1 | Δ |
|---|---|---|
| w/o Conciseness | 46.1 | ↓ 6% |
| w/o Coherence | 42.9 | ↓ 13% |
| w/o Receptiveness | 45.2 | ↓ 8% |
| w/o Pertinence | 46.6 | ↓ 5% |

Table 5: Experimental results (accuracy; %) on the dataset MQuAKE of DeepEdit with KE batch size of 100.

and 52% over MeLLo on the GPT-3.5-Turbo-Instruct, Text-Davinci-003, LLaMA2-7B-Chat respectively over the strong baseline method MeLLo on MQuAKE. On MQuAKE-hard, our DeepEdit outperforms the strong baseline method MeLLo by ×7, ×12, and ×10 times in terms of question answering accuracy with new knowledge on the LLMs LLaMA2-7B-Chat, Text-Davinci-003, and GPT-3.5-Turbo-Instruct respectively.

When the KE batch is full batch, our DeepEdit still outperforms the baseline methods and improves the KE accuracy by a significant margin. Our DeepEdit enhances the LLMs' reasoning on new knowledge through higher conciseness, coherence, pertinence, and receptiveness. The above advantages are especially significant when the LLMs have to utilize multiple new knowledge to get the correct answers because more new knowledge requires LLMs to be correctly aware of the new knowledge on more reasoning steps. The correct knowledge incorporation with our decoding constraints significantly improves the KE performance given more new knowledge. Overall, our DeepEdit achieves substantial improvements on KE for LLMs and consistently outperforms the baseline KE methods on various benchmarks.

### 5.3 Accuracy of Constraint Verification

We conduct an in-depth analysis on the constraint verification of our constraints. We randomly sample 4,000 pairs of sentences from MQuAKE that satisfy our constraints as positive samples and the same number of sentence pairs from different instances as the negative samples. We evaluate different verifiers based on LLaMA2-7B-Chat and GPT-3.5-Turbo-Instruct on the above samples. We report the verification accuracy in Table 4. We observe high verification accuracy of around 80% and over 90% on different constraints achieved by LLaMA2-7B-Chat and GPT-3.5-Turbo-Instruct respectively. This result matches the performance gap between two LLMs on general reasoning tasks (OpenAI, 2022; Touvron et al., 2023).

### 5.4 Efficiency of Depth-first Search

As analyzed in Section 3.3, we utilize the depth-first search to efficiently increase the LLMs' reasoning depth with new knowledge. We evaluate the efficiency of our depth-first search using the 3-hop and 4-hop questions in the MQuAKE-3k benchmark, which holds the longest reasoning chains among the instances.

The methods we evaluate include: MeLLo, the breadth-first search-based decoding method introduced in Sec. 3.3, and our depth-first earch. The average number of reasoning steps and the time of text generation on every instance taken by different methods are reported in Figure 7. where the generation time is on a Linux Server using three A6000 GPUs.

We notice that, compared with the breadth-first search method, our DeepEdit significantly reduces the number of reasoning steps and the generation time. Our DeepEdit efficiently finds the reasoning chain with new knowledge by considering the importance of different step candidates. These results agree with the theoretical analysis in Sec. 3.3, since our DeepEdit' time complexity is $\mathcal{O}(h)$ in the base case, while the time complexity of the breadth-first search is $\mathcal{O}(c^h)$, much higher than our DeepEdit. Compared with the baseline MeLLo, our DeepEdit leads to much less generation time because our output text is purely the reasoning steps without the redundant subquestions and edited facts kept by MeLLo.

### 5.5 Ablation Study

We conduct ablation studies on DeepEdit to empirically examine the contribution of its main technical components. We evaluate the influence of different constraints on DeepEdit. We report the experimental results of the ablation study of constraints in Table 5. We observe that removing each constraint leads to substantial performance degradation in different degrees. This indicates that different constraints contribute to more effective KE from the perspectives of progress, relevance, coherence, and awareness as a whole, and removing any of them will lead to more mistakes in the reasoning of LLMs. The above results also validate that the LLMs' outputs cannot automatically satisfy the constraints that represent the desired properties without our DeepEdit. More experiments and case studies can be found in the appendix.

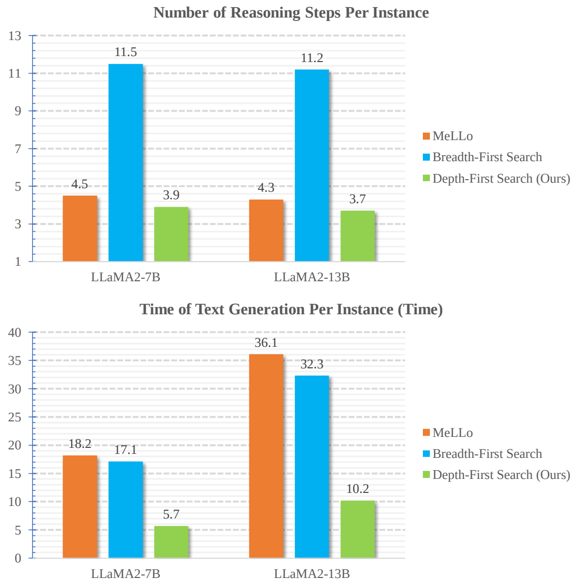


Figure 7: Inference costs of the baseline method MeLLo (Zhong et al., 2023), breadth-first search introduced in Sec. 3.3, and our depth-first search.

## 6 Conclusion

We explore a new paradigm for knowledge editing of black-box LLMs: decoding with constraints. We have designed a new decoding method DeepEdit to improve the knowledge editing of LLMs. DeepEdit enhances LLM's outputs to soundly and efficiently incorporate new knowledge in the multi-step reasoning. In addition to DeepEdit, we provide two new benchmarks: MQuAKE-2002 and MQuAKE-hard to provide more precise and challenging evaluations for knowledge editing methods. Extensive experiments demonstrate the effectiveness of our DeepEdit. Future work includes exploring other constraints that can improve knowledge editing for large language models.

## Acknowledgements

This work was partially supported by a DARPA ANSR program FA8750-23-2-0004. The views and conclusions are those of the authors and should not reflect the official policy or position of DARPA or the U.S. Government.

## References

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2023. Evaluating the ripple effects of knowledge editing in language models. *arXiv preprint arXiv:2307.12976*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. *arXiv preprint arXiv:2210.03329*.

Saibo Geng, Martin Josifosky, Maxime Peyrard, and Robert West. 2023. Flexible grammar-based constrained decoding for language models. *arXiv preprint arXiv:2305.13971*.

Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. 2021. Do language models have beliefs? methods for detecting, updating, and visualizing model beliefs. *arXiv preprint arXiv:2111.13654*.

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

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2020. Gedi: Generative discriminator guided sequence generation. *arXiv preprint arXiv:2009.06367*.

Sachin Kumar, Biswajit Paria, and Yulia Tsvetkov. 2022. Gradient-based constrained sampling from language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2251–2277.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. 2021. Dexperts: Decoding-time controlled text generation with experts and anti-experts. *arXiv preprint arXiv:2105.03023*.

Xin Liu, Muhammad Khalifa, and Lu Wang. 2023. Bolt: Fast energy-based controlled text generation with tunable biases. *arXiv preprint arXiv:2305.12018*.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, et al. 2021. Neurologic a* esque decoding: Constrained text generation with lookahead heuristics. *arXiv preprint arXiv:2112.08726*.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. *arXiv preprint arXiv:2010.12884*.

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

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Tao Meng, Sidi Lu, Nanyun Peng, and Kai-Wei Chang. 2022c. Controllable text generation with neurally-decomposed oracle. *Advances in Neural Information Processing Systems*, 35:28125–28139.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

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

OpenAI. 2022. Introducing chatgpt https://openai.com/blog/chatgpt).

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. *arXiv preprint arXiv:2010.05906*.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. Cold decoding: Energy-based constrained text generation with langevin dynamics. *Advances in Neural Information Processing Systems*, 35:9538–9551.

Matthew Sotoudeh and A Thakur. 2019. Correcting deep neural networks with small, generalizing patches. In *Workshop on safety and robustness in decision making*.

Robert Tarjan. 1972. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Kevin Yang and Dan Klein. 2021. Fudge: Controlled text generation with future discriminators. *arXiv preprint arXiv:2104.05218*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? *arXiv preprint arXiv: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*.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. *arXiv preprint arXiv:2303.11315*.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

## A Analysis on Knowledge Editing Batch Size

We follow the existing work (Zhong et al., 2023) to evaluate the influence of the KE batch sizes on the performance. A larger batch size leads to the more new knowledge collected from more instances for retrieval. We present the test accuracy on MQUAKE-3K with different batch sizes of KE in Figure 8. We observe that the performance of GPT-3.5-TURBO-INSTRUCT with MeLLo and our DEEPEDIT decreases with a larger batch size of knowledge editing. The reason is that with a larger batch, the difficulty of retrieving the relevant new knowledge to the target questions becomes higher. On the other hand, our DEEPEDIT still exhibits significant and consistent improvements over the strong KE method MeLLo.

## B Analysis on the Data where the Edited Facts do not Change the Ground-Truth Answers

Knowledge editing methods enhance the LLMs' reasoning capability on the new knowledge. However, there exists the risk that the KE methods can degrade the LLMs' performance on the data where no new knowledge changes their ground-truth answers. To evaluate whether our method DEEPEDIT degrades the performance of LLMs in this case. We do the following experiments, for every question-answering instance in MQUAKE-3K, we retrieve the edited facts only from other instances as the new knowledge, which would not change the ground-truth answer for the target question. In this setting, we first test the GPT-3.5-TURBO-INSTRUCT without any KE methods, and compare the performance of GPT-3.5-TURBO-INSTRUCT with the KE methods MeLLo and our DEEPEDIT. We present the performance in Table 6.

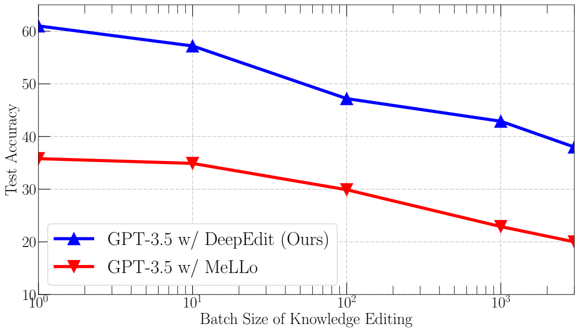


Figure 8: Test accuracy (%) of question answering with new knowledge with different KE batch sizes. We present the performance of MeLLo and our DEEPEDIT applied on GPT-3.5-TURBO-INSTRUCT.

| Method | MQUAKE-3K |
|---|---|
| GPT-3.5-TURBO-INSTRUCT | 55.7 |
| GPT-3.5-TURBO-INSTRUCT w/ MeLLo (Zhong et al., 2023) | 54.2 |
| GPT-3.5-TURBO-INSTRUCT w/ DEEPEDIT (Ours) | **55.5** |

Table 6: Experimental results (accuracy; %) on the dataset MQUAKE-3K with no edited facts change the ground-truth answers.

| Method | COUNTERFACT |
|---|---|
| LLAMA2-7B-CHAT | 85.3 |
| LLAMA2-7B-CHAT w/ MEND (Mitchell et al., 2021) | 57.8 |
| LLAMA2-7B-CHAT w/ MEMIT (Meng et al., 2022b) | 80.9 |
| LLAMA2-7B-CHAT w/ MeLLo (Zhong et al., 2023) | 82.7 |
| LLAMA2-7B-CHAT w/ DEEPEDIT (Ours) | **84.1** |

Table 7: Specificity Score on the dataset COUNTERFACT.

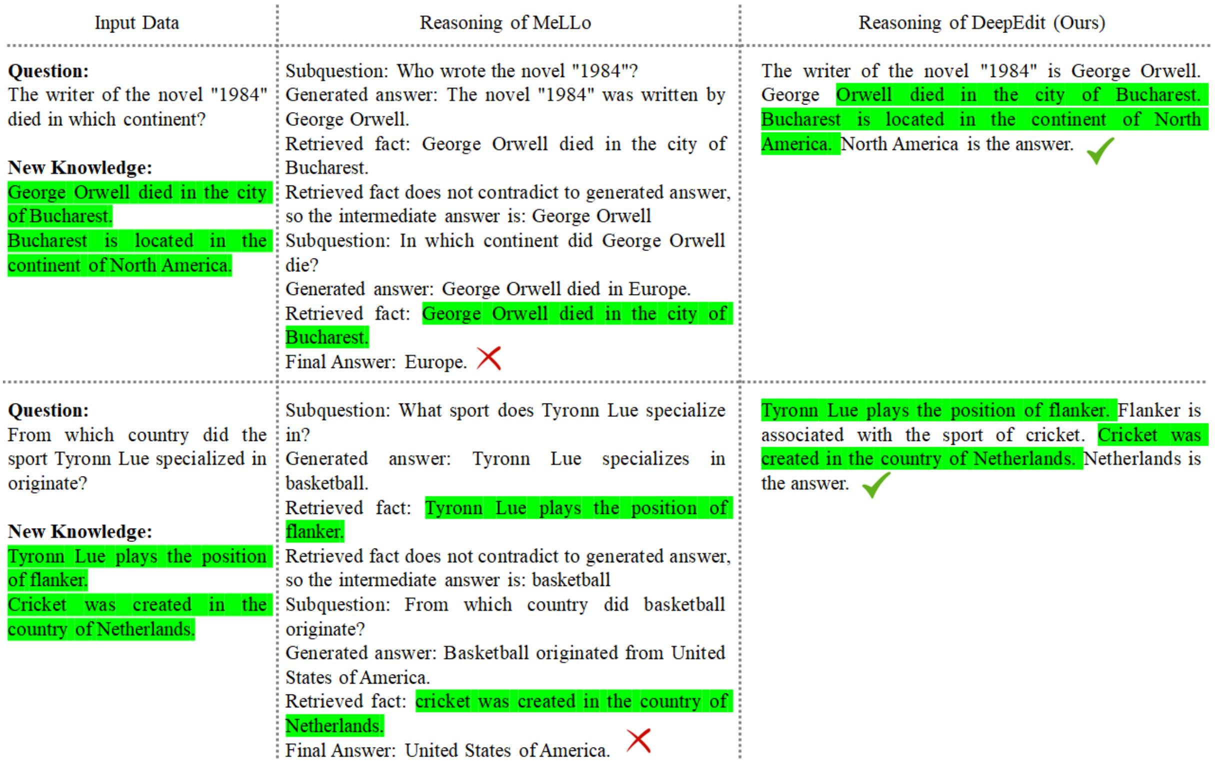
The empirical results show that our DEEPEDIT exhibits higher robustness to give the correct answers for the multi-hop question answering than MeLLo. The performance drop of GPT-3.5-TURBO-INSTRUCT with our DEEPEDIT when no edited facts change the ground-truth labels is negligible.

In addition to the above experiments, we also follow existing work (Meng et al., 2022a; Zheng et al., 2023) to evaluate the Specificity scores of different knowledge editing methods on COUNTERFACT given the KE batch size as full batch, which measures the accuracy of querying neighborhood prompts by Neighborhood Scores. In COUNTERFACT, the neighborhood prompts share the same original object with the target prompt and these facts are not supposed to be edited. The results are presented in Table 7.

Our DEEPEDIT outperforms all the baseline KE methods in terms of the Specificity Score, which validates that our DEEPEDIT causes less performance degradation for LLMs than the baseline methods when the edited facts do not change the ground-truth labels.

## C Case Study

Figure 9 gives a qualitative comparison example between MeLLo and our DEEPEDIT on MQuaKE-3k. The results show that the strong KE method MeLLo makes mistakes on KE. For example, given the question "*The writer of the novel "1984" died in which continent?* ", MeLLo is unable to generate the helpful sub-question to find the final correct answer. In contrast, our DEEPEDIT, effectively



| Input Data | Reasoning of MeLLo | Reasoning of DeepEdit (Ours) |
|---|---|---|
| **Question:**<br>The writer of the novel "1984" died in which continent?<br><br>**New Knowledge:**<br>George Orwell died in the city of Bucharest.<br>Bucharest is located in the continent of North America. | Subquestion: Who wrote the novel "1984"?<br>Generated answer: The novel "1984" was written by George Orwell.<br>Retrieved fact: George Orwell died in the city of Bucharest.<br>Retrieved fact does not contradict to generated answer, so the intermediate answer is: George Orwell<br>Subquestion: In which continent did George Orwell die?<br>Generated answer: George Orwell died in Europe.<br>Retrieved fact: George Orwell died in the city of Bucharest.<br>Final Answer: Europe. ✗ | The writer of the novel "1984" is George Orwell. George Orwell died in the city of Bucharest. Bucharest is located in the continent of North America. North America is the answer. ✓ |
| **Question:**<br>From which country did the sport Tyronn Lue specialized in originate?<br><br>**New Knowledge:**<br>Tyronn Lue plays the position of flanker.<br>Cricket was created in the country of Netherlands. | Subquestion: What sport does Tyronn Lue specialize in?<br>Generated answer: Tyronn Lue specializes in basketball.<br>Retrieved fact: Tyronn Lue plays the position of flanker.<br>Retrieved fact does not contradict to generated answer, so the intermediate answer is: basketball<br>Subquestion: From which country did basketball originate?<br>Generated answer: Basketball originated from United States of America.<br>Retrieved fact: cricket was created in the country of Netherlands.<br>Final Answer: United States of America. ✗ | Tyronn Lue plays the position of flanker. Flanker is associated with the sport of cricket. Cricket was created in the country of Netherlands. Netherlands is the answer. ✓ |

Figure 9: A case study for MeLLo and our DEEPEDIT on the dataset MQuAKE-3k. The new knowledge are in green. We report the prediction of TEXT-DAVINCI-003 with MeLLo and our DEEPEDIT.

controls LLMs to correctly find the relevant new knowledge "Bucharest is located in the continent of North America." and obtain the correct answer North America with respect to the new knowledge.

Taking a closer look, we observe that our DEEPEDIT's induced reasoning is much more succinct than MeLLo, since we effectively filter out the irrelevant and redundant information during decoding. Because longer and noisy textual information can lead to more serious hallucinations and higher difficulty of reasoning, our DEEPEDIT exhibits a more effective KE than MeLLo with its succinct reasoning process controlled by constraints. Overall, our DEEPEDIT enhances LLMs to follow our decoding constraints so as to soundly incorporate new knowledge in multi-step reasoning, which leads to the correct predictions in Figure 9.

Prompt

**Text Generation Prompt of DeepEdit**:

Question: What is the capital of the country where Plainfield Town Hall is located? Thoughts with New Knowledge: Plainfield Town Hall is located in the country of the United States of America.# The capital of United States is Washington, D.C.# Washington, D.C. is the answer. Answer: Washington, D.C.

Question: In which country is the company that created Nissan 200SX located? Thoughts with New Knowledge: Nissan 200SX was created by Nissan.# Nissan is located in the country of Japan.# Japan is the answer. Answer: Japan

Question: Which continent is the country where the director of "My House Husband: Ikaw Na!" was educated located in? Thoughts with New Knowledge: The director of "My House Husband: Ikaw Na!" is Jose Javier Reyes.# Jose Javier Reyes was educated at De La Salle University.# De La Salle University is located in the country of Philippines.# Philippines is located in the continent if Asia.# Asia is the answer. Answer: Asia

Figure 10: Text Generation Prompt of DeepEdit (Zhong et al., 2023)

Prompt

**Text Generation Prompt of MeLLo**:

Question: What is the capital city of the country of citizenship of Ivanka Trump's spouse? Subquestion: Who is Ivanka Trump's spouse? Generated answer: Ivanka Trump's spouse is Jared Kushner. Retrieved fact: Ivanka Biden is married to Samantha Cameron. Retrieved fact does not contradict to generated answer, so the intermediate answer is: Jared Kushner Subquestion: What is the country of citizenship of Jared Kushner? Generated answer: The country of citizenship of Jared Kushner is United States. Retrieved fact: Jared Kushner is a citizen of Canada. Retrieved fact contradicts to generated answer, so the intermediate answer is: Canada Subquestion: What is the capital city of Canada? Generated answer: The capital city of Canada is Ottawa. Retrieved fact: The capital city of United States of America is Seattle. Retrieved fact does not contradict to generated answer, so the intermediate answer is: Ottawa Final answer: Ottawa

Question: Who is the head of state of the country where Rainn Wilson holds a citizenship? Subquestion: What is the country of citizenship of Rainn Wilson? Generated answer: Rainn Wilson is a citizen of United States of America. Retrieved fact: Rainn Wilson is a citizen of Croatia. Retrieved fact contradicts to generated answer, so the intermediate answer is: Croatia Subquestion: What is the name of the current head of state in Croatia? Generated answer: The head of state of Croatia is President Zoran Milanović. Retrieved fact: The name of the current head of state in Croatia is Kolinda Grabar-Kitarović. Retrieved fact contradicts to generated answer, so the intermediate answer is: Kolinda Grabar-Kitarović Final answer: Kolinda Grabar-Kitarović

Question: Who is the spouse of the head of state in United States of America? Subquestion: Who is the head of state in United States of America? Generated answer: The US president is Donald Trump. Retrieved fact: The head of state in United States of America is Joe Biden. Retrieved fact contradicts to generated answer, so the intermediate answer is: Joe Biden Subquestion: Who is the spouse of Joe Biden? Generated answer: The spouse of Joe Biden is Jill Biden. Retrieved fact: The spouse of Joe Bill is Evan Austin. Retrieved fact does not contradict to generated answer, so the intermediate answer is: Jill Biden Final answer: Jill Biden

Question: On which continent is the country of citizenship of the founder of the manufacturer of iPhone 5 situated? Subquestion: Which company is iPhone 5 produced by? Generated answer: iPhone 5 is produced by Apple. Retrieved fact: The company that produced iPhone 5 is Iveco. Retrieved fact contradicts to generated answer, so the intermediate answer is: Iveco Subquestion: Who is the founder of Iveco? Generated answer: Iveco was founded by Giovanni Agnelli. Retrieved fact: House of Bonaparte was founded by Gustav I of Sweden. Retrieved fact does not contradict to generated answer, so the intermediate answer is: Giovanni Agnelli Subquestion: What is the country of citizenship of Giovanni Agnelli? Generated answer: Giovanni Agnelli is a citizen of Italy. Retrieved fact: Giovanni Agnelli is a citizen of Niger. Retrieved fact contradicts to generated answer, so the intermediate answer is: Niger. Subquestion: On which continent is Niger situated? Generated answer: Niger is situated on Africa. Retrieved fact: Kingdom of England is located in the continent of North America. Retrieved fact does not contradict to generated answer, so the intermediate answer is: Africa. Final answer: Africa

Figure 11: Text Generation Prompt of MeLLo (Zhong et al., 2023).