Knowledge Editing through Chain-of-Thought

Anonymous ACL submission

Abstract

Knowledge Editing is a technique that updates large language models (LLMs) with new information to maintain their world knowledge. This approach avoids the need to rebuild the model from scratch, thereby addressing the high costs associated with frequent retraining. Among these, the in-context editing paradigm stands out for its effectiveness in integrating new knowledge while preserving the model's original capabilities. Despite its potential, existing in-context knowledge editing methods are often task-specific, focusing primarily on multi-hop QA tasks using structured knowledge triples. Moreover, their reliance on few-shot prompting for task decomposition makes them unstable and less effective in generalizing across diverse tasks. In response to these limitations, we propose **EditCoT**, a novel knowledge editing framework that flexibly and efficiently updates LLMs across various tasks without retraining. EditCoT works by generating a chain-of-thought (CoT) for a given input and then iteratively refining this CoT process using a CoT editor based on updated knowledge. We evaluate EditCoT across a diverse range of benchmarks, covering multiple languages and tasks. The results demonstrate that our approach achieves state-of-the-art performance while offering superior generalization, effectiveness, and stability compared to existing methods, marking a significant advancement in the field of knowledge updating[1].

1 Introduction

Large Language Models (LLMs) have shown remarkable performance across numerous NLP tasks in recent years. However, once an LLM has been fully trained, its parametric knowledge becomes fixed and constrained. Retraining the entire model to incorporate new information is both expensive and time-consuming. To address this challenge, the field of knowledge editing has emerged, offering techniques to modify or add knowledge efficiently while preserving most of the original model's parameters. Existing knowledge edit approaches can be broadly categorized into two types: parametric and non-parametric (Wang et al., 2024b). Parametric methods directly modify the model's weights to integrate new knowledge. While this approach enables precise editing, existing research indicates that such changes may negatively affect the model's original performance and interfere with previously learned knowledge that should remain unchanged (Gu et al., 2024b). In contrast, non-parametric methods, such as in-context editing, are increasingly popular because they utilize the in-context learning ability of LLMs without modifying the original model weights, thereby preserving the model's foundational capabilities and existing knowledge (Zheng et al., 2023).

Most existing in-context knowledge editing frameworks are designed for multi-hop question answering (MQA) tasks using knowledge triples as the primary knowledge representation. Researchers construct multi-hop questions by linking multiple triples to evaluate these methods' performance (Zhong et al., 2023). For instance, Mello and PokeMQA guide LLMs in task decomposition with manually crafted examples, facilitating knowledge updates for sub-questions (Zhong et al., 2023; Gu et al., 2024a). RAE (Shi et al., 2024b) constructs and then edits a knowledge graph, utilizing the retrieval and pruning of the knowledge graph to obtain contextually injected knowledge.

While effective for triple-based MQA tasks, these methods are overly specialized and struggle with other tasks and complex reasoning. For example, RAE is impractical for the LeKUBE(Wang et al., 2024b) legal dataset, where constructing a knowledge graph on its corpus is unfeasible. PokeMQA's strategy of using updated knowledge directly as answers to sub-questions works well

---

[1] Code and data available at: https://anonymous.4open.science/r/EditCoT-6E40

for triple-based tasks but fails with more complex tasks like DUNE (Akyürek et al., 2023). Furthermore, Mello and PokeMQA's reliance on few-shot prompting for task decomposition results in instability and reduce effectiveness across diverse tasks.

To address the limitations of existing in-context editing methods, we propose **EditCoT**, a framework that edits LLM's knowledge through chain-of-thought (CoT). EditCoT is also an in-context editing method that does not change the parameters of LLMs used for reasoning, thereby preserving the original capacity of the backbone LLMs. Unlike previous methods that are task-specific or heavily reliant on knowledge triples, EditCoT iteratively refines the model's reasoning with retrieved updated knowledge. This flexible approach allows it to dynamically adapt its reasoning without requiring task-specific adjustments or predefined knowledge structures. Specifically, EditCoT starts by instructing the LLM to first generate an answer to the question, and then construct an initial CoT based on the question and answer. The original CoT is then iteratively refined by a CoT editor, which requires only a single training session without needing retraining when encountering new knowledge. The final answer is generated by prompting the LLM to inference based on the updated CoT.

We evaluate EditCoT on a variety of benchmarks, including the triple-based multi-hop QA dataset MQuAKE-CF-3k(Zhong et al., 2023), a free-form editing task dataset DUNE(Akyürek et al., 2023), and a domain-specific Chinese legal knowledge update dataset LeKUBE(Wang et al., 2024b). They cover different editing formats and tasks in both Chinese and English. Experimental results demonstrate that our method achieves state-of-the-art performance with better efficiency and robustness.

The contributions of this paper are as follows:

- We propose a novel framework, *EditCoT*, that edits the chain-of-thought of LLMs, enabling more effective and task-agnostic knowledge updates.
- We demonstrate our method performs well across various knowledge editing benchmarks with different editing formats and task types, exhibiting significantly better effectiveness, robustness, and generalizability compared to other baselines.

2 Related Work

2.1 Knowledge Editing

Knowledge editing methods modify or update knowledge within LLMs, categorized into parametric and non-parametric approaches (Wang et al., 2024b). ROME (Meng et al., 2022) is a parametric method, update the model's parameters by treating FFN layers as key-value storage for precise modifications. Non-parametric approaches leverage in-context learning (Brown et al., 2020), such as Mello (Zhong et al., 2023), which uses prompts for task decomposition, PokeMQA (Gu et al., 2024a), which enhances Mello's robustness with a scope detector, and RAE (Shi et al., 2024b), which uses a knowledge graph for task-specific editing.

2.2 Chain-of-Thought

Chain-of-thought (CoT) has significantly enhanced LLMs by providing step-by-step reasoning for complex tasks. Wei et al. (2022) introduce CoT prompting, where manually supplied reasoning chains help LLMs generate multi-step solutions. Kojima et al. (2022) demonstrate phrases like "Let's think step by step" enable zero-shot CoT reasoning. Recent work has focused on enhancing the factuality of CoT. Wang et al. (2023b) propose a self-consistency decoding method to improve CoT reliability. Zhao et al. (2023) introduce Verify-and-Edit that involves editing the generated CoT. However, our approach significantly differs in some key aspects. The verification process of Verify-and-Edit relies on self-consistency, which is less effective for knowledge editing due to the LLM's high confidence in outdated information. Additionally, their method depends on the LLM's ability to learn from context, whereas our CoT editor provides more comprehensive adjustments. Further comparative analysis can be found in Appendix B.

3 Methodology

In this section, we present the methodology of the EditCoT framework in detail. We begin with an overview of our proposed approach, followed by a step-by-step description of the framework's workflow. We then explain the CoT editor training process in depth, concluding with a discussion of implementation details and dataset construction.

3.1 Methodology Overview

EditCoT aims to update LLMs by editing their CoT, integrating new factual knowledge while preserving their reasoning capabilities. Traditional parametric knowledge editing methods often require resource-intensive retraining or fine-tuning, which can result in the loss of existing knowledge. In contrast, EditCoT introduces an iterative editing

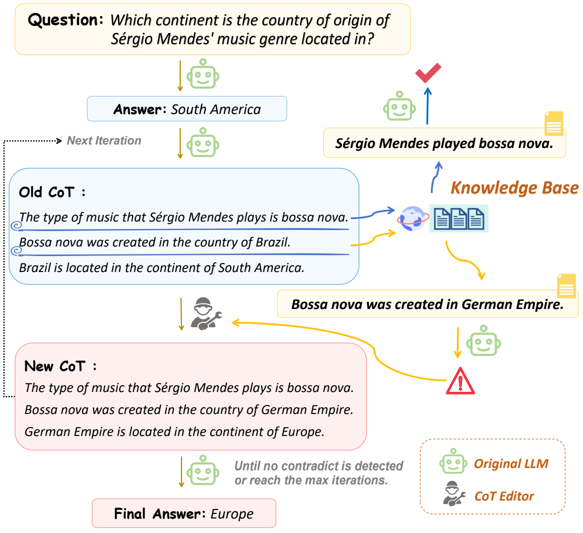

Question: Which continent is the country of origin of Sérgio Mendes' music genre located in?
Answer: South America
Next Iteration
Sérgio Mendes played bossa nova.
Knowledge Base
Old CoT :
The type of music that Sérgio Mendes plays is bossa nova.
Bossa nova was created in the country of Brazil.
Brazil is located in the continent of South America.
Bossa nova was created in German Empire.
New CoT :
The type of music that Sérgio Mendes plays is bossa nova.
Bossa nova was created in the country of German Empire.
German Empire is located in the continent of Europe.
Until no contradict is detected or reach the max iterations.
Original LLM
CoT Editor
Final Answer: Europe

Figure 1: EditCoT updates LLMs by iteratively editing the CoT to integrate new knowledge while preserving its reasoning abilities. It generates an initial CoT, detects conflicts, and revises the CoT iteratively.

mechanism that efficiently incorporates updates by focusing solely on the CoT, ensuring that the model remains both accurate and logically consistent.

The framework leverages two components: an unedited version of the LLM responsible for generating the initial CoT, detecting conflict, and generating the final answer, and an editor capable of editing CoT based on the new knowledge. The following subsection details each step of EditCoT.

3.2 Workflow of EditCoT

Figure 1 illustrates the EditCoT framework's overall workflow, which is also described in pseudocode in Appendix A. The core steps are as follows:

Step 1: Generate Initial CoT by the Original LLM. At the beginning of the process, the unedited LLM generates an initial CoT in response to a query. This CoT outlines the sequence of reasoning steps leading to the model's final answer. Prioritizing the completeness of the reasoning path over the answer's accuracy, we first let the model provide the answer directly and then generate the CoT based on the question and answer.

Step 2: Retrieve Step by Step and Detect Conflicts. Once the initial CoT is generated, we treat each step of the CoT as a query and retrieve it from the memory of edited knowledge. This memory is typically organized in two ways: one where the pre- and post-edit knowledge are paired together (e.g., MQuAKE (Zhong et al., 2023)), and another where only the post-edit knowledge is provided (e.g., DUNE (Akyürek et al., 2023)). For the former, we search the knowledge base for the pre-edit knowledge and then match the post-edit knowledge according to the corresponding relationship. For the latter, we directly search within the post-edit knowledge. Once updated knowledge is found, we proceed to the conflict detection stage.

EditCoT utilizes the original LLM to detect knowledge conflicts by engaging in a three-class classification task: conflict, support, or irrelevant. Specifically, given a question, the initial CoT, and the retrieved knowledge, the classification is determined based on the probability of the prefix of the corresponding word output by the LLM. This process is crucial for identifying whether the original reasoning diverges from updated knowledge.

Step 3: Edit the chain-of-thought Once a knowledge conflict is detected, we input the question, new knowledge, and old CoT into a CoT editor, which generates a new CoT. Then, our iterative process returns to Step 2, where this new CoT is treated as the old CoT for the next iteration. Each iteration resolves one edit at a time, progressively addressing multiple potential conflicts through iterations. The process stops when all conflicts are found or the iteration limit is reached.

Step 4: Get Final Answer Once the CoT has been fully edited and no further conflicts are detected, the final chain-of-thought is used to generate the final answer. The original LLM is responsible for this step, ensuring that the model's answer adheres to its existing inferential abilities, while also incorporating the new factual knowledge. We instruct the original LLM to pay more attention to the new CoT in the context since the new CoT often conflicts with the model's intrinsic knowledge.

3.3 CoT Editor Training

We design an automated approach to generate training data required for the CoT editor, where human annotation is unnecessary, except for the use of some human-posed questions. Our method does not rely on state-of-the-art LLMs like GPT-4(OpenAI et al., 2023) and is not tailored to specific datasets or tasks. The editor learns to edit CoT during training and can apply this skill to various tasks without retraining when encountering new knowledge. Additionally, the CoT editor is trained from the original LLM, ensuring that no knowledge distillation issues arise.

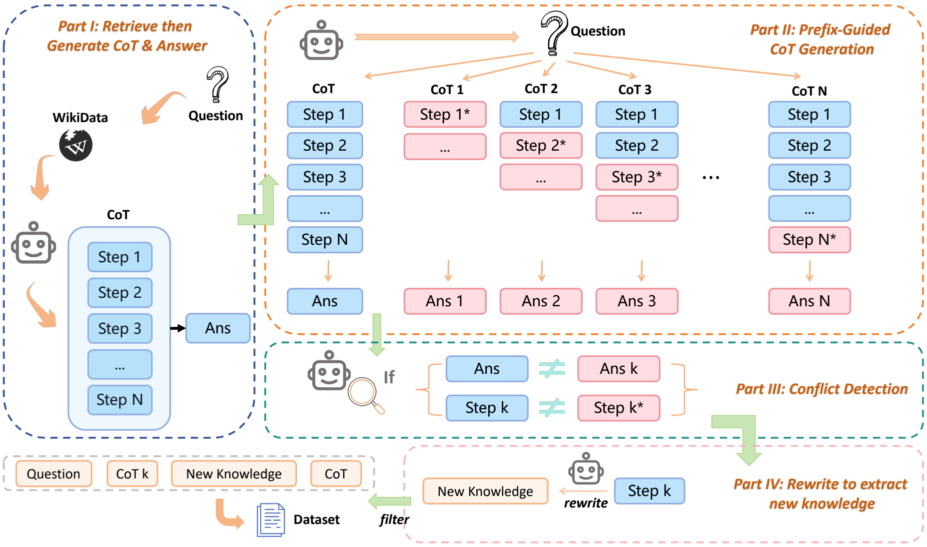


Figure 2: Diagram of the dataset construction: Our data source consists solely of questions from HotpotQA, without requiring human-annotated answers or related articles. In Part II, the blue sections represent the prefix used to guide generation, which have been generated in Part I, while red sections indicate newly generated reasoning paths.

3.3.1 Task Formulation

The editor's input includes a question Q, an original CoT, and new knowledge K_{new} that conflicts with the CoT. The goal is to generate a revised chain-of-thought CoT_{new} that incorporates new knowledge. Formally, this can be expressed as:

$$\text{Editor}(Q, \text{CoT}, K_{\text{new}}) \rightarrow \text{CoT}_{\text{new}},$$

3.3.2 Data Construction

To construct training data, we start by collecting high-quality, human-generated questions. In practice, we use the validation and test sets from the HotpotQA(Yang et al., 2018) dataset, which contains diverse multi-hop questions that often require multiple Wikipedia articles to answer. HotpotQA is selected for its diversity of question types, and multi-hop reasoning tasks generally yield more complex CoTs. After obtaining these questions, we use the following construction steps:

Part I: Retrieve, then Generate CoT and Answer For each question, we first retrieve top-5 relevant Wikipedia paragraphs and then use a retrieval-augmented generation (RAG) method to generate the CoT and the corresponding answer. We consider this initial CoT and answer as a basic version, and in the following sections, these are highlighted as CoT and Answer for clarity. To facilitate step-level separation, we prompt the model to indicate each step with the `[STEP]` phrase.

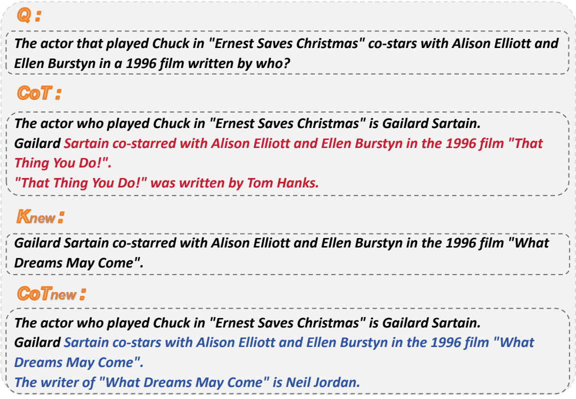
Part II: Prefix-Guided CoT Generation Given a CoT with N steps generated in Part I, for each step, e.g. the kth step ($k = 1, 2, \ldots, N$), we take the first $(k-1)$ steps as a prefix and prompt the model to generate the remaining steps and the final answer, marked as CoT k and Answer k respectively. It is important to note that this step does not use RAG, and the model is only prompted with the first $(k-1)$ steps of the CoT as context. We finally obtain N CoTs in this part.

Part III: Conflict Detection We then compare each of the N CoTs generated in Part II to the original CoT and instruct the model to detect any knowledge conflicts. A data point is accepted into the final dataset only if: (1) there is a conflict in the final answers, and (2) there is a conflict at the first diverging step between the two CoTs.

For example, considering CoT k and CoT, the first diverging step is step k. If the original Step k and the new Step k conflict with each other, and there is also a conflict between Answer and Answer k, they are eligible to be added to the final dataset. This ensures that conflicting knowledge can be pinpointed at a specific step in the reasoning chain. Here we assume that the divergence occurs only once in most of the cases.

Part IV: Rewrite to Extract New Knowledge In this step, we rewrite the identified diverging step from the original CoT to explicitly extract the new knowledge. Specifically, for the example above, a complete training data entry is formed as a tuple:

$$(Q,\ \text{CoT},\ K_{\text{new}},\ \text{CoT}_{\text{new}}) =$$
$$(\text{Question},\ \text{CoT k},\ \text{rewrite}(\text{Step k}),\ \text{CoT}).$$



Q :

The actor that played Chuck in "Ernest Saves Christmas" co-stars with Alison Elliott and Ellen Burstyn in a 1996 film written by who?

CoT :

The actor who played Chuck in "Ernest Saves Christmas" is Gailard Sartain.
Gailard Sartain co-starred with Alison Elliott and Ellen Burstyn in the 1996 film "That Thing You Do!".
"That Thing You Do!" was written by Tom Hanks.

K_{new} :

Gailard Sartain co-starred with Alison Elliott and Ellen Burstyn in the 1996 film "What Dreams May Come".

CoT_{new} :

The actor who played Chuck in "Ernest Saves Christmas" is Gailard Sartain.
Gailard Sartain co-stars with Alison Elliott and Ellen Burstyn in the 1996 film "What Dreams May Come".
The writer of "What Dreams May Come" is Neil Jordan.

Figure 3: A training data example. The black part of the two CoTs indicates a shared prefix.

We also apply regular expressions to filter out low-quality examples, such as those overly influenced by the retrieved documents in RAG. Figure 2 illustrates the overall process of dataset construction.

3.4 Implementation Details for Training

We implement conflict detection using a multiple-choice format. If the model assigns a probability of 0.6 or higher to a conflict between two answers or reasoning steps, we consider them inconsistent. During prefix-guided generation, to encourage the LLM to focus on the same topic at the diverging step, we provide the first word of the original CoT at the diverging step to the LLM. Prompt templates for dataset construction are detailed in subsection J.1. The dev and test sets of HotpotQA contain 14,810 questions. Using Meta-Llama-3-8B-Instruct(Dubey et al., 2024) and Qwen2.5-14B-Instruct(Yang et al., 2024; Team, 2024), we generate 459 and 316 training examples, respectively. Figure 3 illustrates a constructed data instance. During training, we calculate the loss based on CoT_{new} using cross-entropy loss.

4 Experimental Settings

4.1 Dataset and Metrics

We evaluate the performance of EditCoT and other baselines across the following three datasets:

MQuAKE-CF-3k(Zhong et al., 2023) This dataset is designed for knowledge editing using Wikipedia-based knowledge triples, with queries in the form of multi-hop questions. Each question involves 1-4 edits. The evaluation metric is *Multihop-Accuracy*, as in the original paper: each instance contains 3 questions, and if anyone is answered correctly, it is considered accurate. We follow the settings of Mello and PokeMQA, using *Exact Match* to assess correctness.

DUNE(Akyürek et al., 2023) It includes various editing tasks, where edits are natural language instructions for the model to follow specific facts or requirements. It covers correcting errors, adding new information, etc. We conduct evaluations on 3 subsets: *Arithmetic Reasoning*, *Scientific Reasoning*, and *New Information*. The metric is *Accuracy*.

LeKUBE(Wang et al., 2024a) A Chinese legal knowledge editing dataset . We focus on two challenging tasks, where prior baselines show poor performance: *Multiple-Choice Questions of the Legal Scenario* and *True/False Questions of Change in Statute*. The metric is *Accuracy*. The statistics of the three benchmarks are detailed in Appendix D.

4.2 Models

We evaluate two open-source models: Meta-Llama-3-8B-Instruct(Dubey et al., 2024) and Qwen2.5-14B-Instruct(Team, 2024). Since the former does not support Chinese, we evaluate the latter on LeKUBE. Following LeKUBE's protocol, we first fine-tune the model on the STARD corpus(Su et al., 2024a), which contains Chinese laws and judicial interpretations, to ensure the model acquires Chinese legal knowledge before any updates.

We also assess GPT-4o-2024-08-06(OpenAI et al., 2023) on MQuAKE-3k-CF. Due to its proprietary nature, we test only a few non-parametric editing methods on this model. Following RAE(Shi et al., 2024b), we randomly sample 300 data points to minimize costs. Meta-Llama-3-8B-Instruct is used as a proxy model for tasks the proprietary model cannot perform. In RAE, this proxy extracts and prunes knowledge graphs based on probabilities, while in EditCoT, it acts as a CoT editor.

4.3 Baselines

We compare EditCoT with a range of model editing techniques, including knowledge neuron-based (KN)(Dai et al., 2022), rank-one model editing (ROME)(Meng et al., 2022), and in-context methods such as Mello(Zhong et al., 2023), PokeMQA(Gu et al., 2024a), and RAE(Shi et al., 2024b). We also include full-parameter fine-tuning (FT) and retrieval-augmented generation (Naive RAG)(Lewis et al., 2020) as baselines. To distinguish our approach from the RAG method, we evaluate two advanced variants, RAT(Wang et al., 2024c) and FLARE(Jiang et al., 2023), on the

MQuAKE and DUNE benchmarks. RAT incorporates multi-turn query rewriting, retrieval, and CoT enhancements to address complex tasks effectively. FLARE dynamically determines the necessity of retrieval during model inference. These comparisons highlight the superiority of our approach in performing knowledge editing tasks.

Notably, since LeKUBE is a Chinese dataset, we translate the official prompts for Mello and PokeMQA into Chinese to accommodate their sensitivity to language type. RAT and FLARE are not evaluated on LeKUBE because it is challenging to align their settings with those of the dataset. RAE is not applicable to datasets other than MQuAKE-CF-3k since it requires construction of knowledge graphs, which is impractical on these datasets. Details on the baselines are available in Appendix E.

4.4 Implementation Details

For the CoT editor, the data construction process is outlined in subsection 3.4. We use 80% of the filtered data generated by the respective model and train it for 1 epoch. To generate the initial answer, CoT, and final answer, and detect conflicts, we use 5-shot prompts styled after Mello and PokeMQA and apply this format across all tasks, as detailed in subsection J.2 . For conflict detection, we instruct the model to classify outputs into three categories: "Contradict", "Support", or "Unrelated". We take the token with the highest probability as the final classification. While we use the `[STEP]` token during dataset construction, in actual inference, we allow the model to freely generate the CoT, with steps separated by newline characters.

For the EditCoT retrieval corpus, settings are provided in section 3.2. For datasets offering both pre- and post-edit knowledge (e.g., MQuAKE-CF-3k and LeKUBE), we retrieve from the pre-edit corpus and match with corresponding post-edit sentences. For datasets with only post-edit knowledge (e.g., DUNE), we directly retrieve from the post-edit corpus. We use the same retriever, Contriever(Izacard et al., 2022), as Mello in MQuAKE-CF and DUNE. For Chinese datasets, we use BM25(Robertson et al., 2009), which performs well in LeKUBE.

5 Experimental Results

5.1 Effectiveness

Generalizability. EditCoT demonstrates superior dataset generalization and model robustness compared to baselines. As shown in Table 1, EditCoT achieves top performance on 6 out of 8 results and second-best on 2 out of 8 results in MQuAKE-3k-CF and DUNE, outperforming all baselines. It consistently excels across two LLMs, unlike other methods like Mello, PokeMQA, and RAE, which perform well on Llama-3-8B-Instruct but poorly on Qwen2.5-14B-Instruct. Mello and PokeMQA's lack of robustness stems from their dependence on strict instructions and hand-crafted few-shot examples, which exhibit varying effectiveness across models and tasks. Additionally, safety-aligned models often reject external edits. To provide a detailed comparison between our method and the baselines, we include case studies in Appendix I.

Applicability to Vertical Domains and Black-box Models. EditCoT is also effective in specialized vertical domains and proprietary models. Table 2 presents the evaluation results on LeKUBE, where EditCoT outperforms all baselines across two tasks, significantly surpassing other methods in the multiple-choice questions of the legal scenario. Furthermore, Table 3 reports GPT-4o's performance on MQuAKE-CF, where EditCoT also demonstrates competitive results, achieving second-best overall. Notably, the proxy model in this experiment is Meta-Llama-3-8B-Instruct. When we compare the performance of the proxy model itself with the GPT-4o, we find that, while RAE's accuracy improves from 54.1% to 59.7% with GPT-4o, EditCoT achieves a larger relative improvement, rising from 35.4% to 45.0%. This suggests that EditCoT relies more on the tested LLM's internal reasoning, while the CoT Editor (8B size) emphasizes editing over reasoning. And RAE benefits primarily from leveraging external knowledge.

Challenges in General-Purpose In-Context Editing. Developing a broadly applicable in-context editing framework remains a significant challenge across both datasets and models. The three datasets span multihop reasoning, generalized edits, domain-specific knowledge, and different languages, posing difficulty for current editing methods. Among the baselines, PokeMQA performs well on multi-hop questions with Llama-3, and simpler methods like Naive RAG excel in narrow settings. However, these methods perform poorly on other datasets or models. RAE excels on the MQuAKE-CF-3k, because it constructs structured knowledge graphs and accesses a broader range of external knowledge (e.g., Wikipedia), but fails to generalize beyond structured datasets. In contrast, EditCoT addresses these limitations, achieving the

Table 1: Results on MQuAKE-CF-3k and three subsets of DUNE, with MQuAKE-CF-3k using the Multi-hop Accuracy metric and the others using Accuracy. The best performance for each dataset and model is highlighted in bold, and the second-best is underlined. A dash ("-") indicates that the method is not applicable to the given dataset. Llama and Qwen represent Llama-3-8B-Instruct and Qwen2.5-14B-Instruct, respectively.

Models	**Datasets**		**Parametric Methods**			**RAG Methods**			**In-Context Editing**			
			FT	**KN**	**ROME**	**Naive RAG**	**RAT**	**FLARE**	**Mello**	**PokeMQA**	**RAE**	**EditCoT(Ours)**
Llama	**MQuAKE-CF-3k**		11.2	2.8	4.0	9.7	0.7	3.3	10.0	26.1	**54.1**	35.4
	DUNE	**Arithmetic**	74.1	80.4	80.5	84.0	70.0	40.1	73.7	83.8	-	**90.7**
		Scientific	11.1	82.1	81.4	81.6	84.3	82.1	77.1	61.7	-	**85.0**
		New Info	30.3	66.6	69.1	89.2	77.0	81.6	89.8	58.7	-	**91.3**
Qwen	**MQuAKE-CF-3k**		9.3	3.2	0.0	10.0	3.3	3.7	5.8	5.3	26.9	**34.2**
	DUNE	**Arithmetic**	82.8	86.9	86.9	89.7	82.6	23.4	61.0	35.2	-	**97.3**
		Scientific	13.8	75.3	75.5	70.7	86.0	83.2	37.0	7.7	-	**86.3**
		New Info	76.2	73.4	73.1	**96.8**	87.4	70.1	34.3	17.3	-	93.8

Table 2: Applicability to vertical domains: LeKUBE is a Chinese Legal Knowledge Editing Benchmark.

Model	**Subsets of LeKUBE**	**Fine-tuning**	**Naive RAG**	**KN**	**ROME**	**Mello**	**PokeMQA**	**EditCoT(Ours)**
Qwen2.5-14B-Instruct	**MCQ of the Legal Scenario**	42.2	47.2	38.9	38.3	6.7	0.0	**58.3**
	T/F Questions of Change in Statute	65.9	69.2	55.8	56.1	18.6	0.0	**69.5**

Table 3: Results on GPT-4o (a black-box LLM). Llama-3-8B-Instruct serves as a proxy LLM for RAE and EditCoT. The dataset is 300 samples from MQuAKE-CF.

	Naive RAG	**Mello**	**PokeMQA**	**RAE**	**EditCoT(Ours)**
Score	12.0	15.0	9.7	**59.7**	45.0

Table 4: Locality test on the New Info (Locality) set of DUNE. EditCoT is the best in all methods.

	Before Editing	**Naive RAG**	**Mello**	**PokeMQA**	**EditCoT**
Accuracy	65.2	34.3	58.0	45.6	**59.7**

best or second-best performance across datasets and models, although not always optimal.

RAG methods underperform compared to EditCoT. There are two reasons: 1) RAG methods seldom address conflicts between parametric and contextual knowledge, which are common in knowledge editing. While RAT modifies CoT, Table 5 shows that EditCoT, using a trained editor, injects knowledge more effectively than a simple prompt. 2) Certain assumptions of RAG are incompatible with knowledge editing. FLARE dynamically decides on retrieval based on model uncertainty about generated words. However, in knowledge editing, the LLM is often confident about the outdated knowledge. In Appendix C, we provide a more detailed discussion on the differences and advantages of EditCoT relative to the RAG methods.

5.2 Robustness

In this section, we evaluate EditCoT against other in-context knowledge editing methods, using Llama3-8B-Instruct from multiple perspectives.

Locality: Locality in knowledge editing assesses a method's ability to update specific knowledge without impairing untargeted information (Mitchell et al., 2022). We evaluate locality using the New Information (Locality) subset of DUNE, where the retrieval scope is limited to the edited set (consistent with the main experimental setup), but test tasks pertain to knowledge outside this set. The results, presented in Table 4, are compared with the unedited model's performance. While all editing methods show some decline in performance relative to the unedited model, EditCoT exhibits the best locality. RAG performs the worst, as it lacks conflict detection between documents and queries, resulting in poor handling of irrelevant information.

Performance with Different Batch Sizes: The performance of knowledge editing methods varies with the editing batch size. Here we denote the batch size as the number of questions in one batch. We test the sensitivity of Mello, PokeMQA, and EditCoT to batch sizes.[2] We evaluate batch sizes of 1, 10, 100, and 300 on a 300-sample dataset from MQuAKE-CF-3k. Results in Figure 5 indicate that all methods perform best with a single sample, with performance decreasing as size increases. However, EditCoT consistently outperforms the others across all batch sizes, showing the smallest performance degradation (around 20%) from size 1 to 300, compared to declines of over 30% for the others.

---

[2] We don't compare RAE because the locality of in-context editing is influenced by retrieval. RAE uses knowledge graph retrieval, which is not comparable to the other three methods.

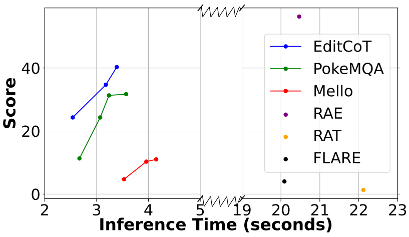


Figure 4: Inference time vs. performance: Top-left indicates shorter inference time and higher performance. Figure split due to wide horizontal span.

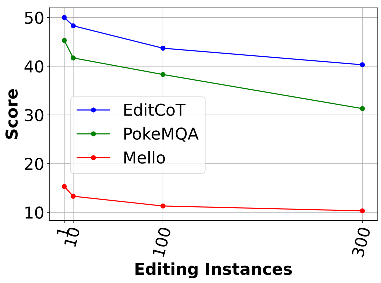


Figure 5: Performance with different batch sizes on MQuAKE-CF. The metrics here is multi-hop accuracy.

5.3 Efficiency

We investigate efficiency on 100 sampled questions from MQuAKE-CF-3k. By varying the max-iterations for Mello, PokeMQA, and EditCoT, we record their corresponding inference time and performance. Complete experimental details are in Appendix F. As shown in Figure 4, EditCoT achieves superior performance with the shortest inference time, outperforming baselines at similar computational costs. The efficiency advantage stems from requiring fewer iterations (3.65 average iterations for EditCoT vs. 4.33 average iterations for PokeMQA) and demonstrating stronger early performance within initial iterations. Further details on the impact of iteration limits are in Figure G. The average inference time for RAE and two advanced RAG methods exceeds 20 seconds, making their application in real-world tasks challenging.

5.4 Ablation Study

In this section, MQuAKE-CF means a dataset of 300 examples sampled from MQuAKE-CF-3k.

Impact of Components We evaluate two components of our method, as shown in Table 5. Removing conflict detection before editing the CoT or employing the original model as the CoT editor without specialized training result in considerable performance deterioration. Additionally, the degree of performance decline varies among different datasets when one component is removed. These results demonstrate the critical role of both components: conflict detection prevents contamination from irrelevant retrieved information, while training the editor enables effective utilization of model editing capabilities beyond simple prompting. Furthermore, in Appendix H, we examine whether training the CoT editor results in data leakage, ensuring the fairness of the experiment.

Table 5: Impact of conflict detection and trained CoT editor. L-8B represents Llama-3-8B-Instruct and Q-14B represents Qwen2.5-14B-Instruct.

	MQuAKE-CF		MCQ in LeKUBE
	L-8B	**Q-14B**	**Q-14B**
EditCoT	40.3	40.3	58.3
EditCoT w/o conflict detection	32.3	33.0	48.9
EditCoT w/o trained CoT editor	29.7	26.7	52.2

Table 6: Impact of training settings on the performance of the CoT editor, evaluated on MQuAKE-CF.

# Epoch	**1**	**2**	**3**	**4**	**5**
Accuracy	40.3	**40.7**	40.3	40.0	40.0
# Training Data	**50**	**100**	**200**	**250**	**316(All)**
Accuracy	39.0	39.3	**40.7**	40.3	40.3

Impact of Training Configurations Table 6 illustrates the effect of training dataset sizes and epochs on the CoT editor. The model tested is Qwen2.5-14B-Instruct. While increasing the number of epochs has little impact, performance stabilizes after expanding the training size to 200 examples, suggesting that the CoT editor requires a sufficient amount of diverse data to learn effectively. This supports our main experimental setup, where training for one epoch with 80% of generated data (252 examples) is a reasonable choice.

6 Conclusion

In this paper, we introduce **EditCoT**, a novel framework designed for efficiently updating LLMs through iterative CoT editing. EditCoT enhances the generalizability and robustness of knowledge editing across diverse tasks, without the need for retraining the LLM each time new knowledge is encountered. Our experiments, conducted on a variety of benchmarks, demonstrate that EditCoT consistently outperforms existing methods, proving its effectiveness in both general-purpose and domain-specific applications. This approach presents a flexible and efficient solution for continuous knowledge updates in LLMs, marking a significant advancement in the field of knowledge editing.

7 Limitations

EditCoT has certain limitations, particularly in its reliance on two distinct models. While this design choice does not increase inference time, it does result in higher GPU resource consumption, which can be challenging in environments with limited hardware resources. However, in practical applications, there is always a trade-off between performance and resource consumption. Compared to other in-context editing approaches, EditCoT offers notable improvements in both efficiency and performance (as shown in Figure 4). As for resource consumption, the experiments in Table 3 demonstrate that even with large-scale models like GPT-4o, employing an 8B CoT editor as a proxy model delivers strong performance while incurring minimal resource overhead relative to the base LLM being edited (GPT-4o). This implies that in real-world applications, the CoT Editor can be much smaller than the original LLM.

For the CoT editor, the extent to which the model size can be reduced and the potential for further compression have not been fully explored in this work. Future work could explore the possibility of smaller editors, aiming to reduce the computational burden without compromising performance. This could make EditCoT more accessible and practical for a wider range of applications and deployment scenarios.

References

Afra Akyürek, Eric Pan, Garry Kuwanto, and Derry Wijaya. 2023. DUnE: Dataset for unified editing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1847–1861, Singapore. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Junjie Chen, Weihang Su, Zhumin Chu, Haitao Li, Qinyao Ai, Yiqun Liu, Min Zhang, and Shaoping Ma. 2024. An automatic and cost-efficient peer-review framework for language generation evaluation. *arXiv preprint arXiv:2410.12265*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar

Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models.

Hengrui Gu, Kaixiong Zhou, Xiaotian Han, Ninghao Liu, Ruobing Wang, and Xin Wang. 2024a. PokeMQA: Programmable knowledge editing for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8069–8083, Bangkok, Thailand. Association for Computational Linguistics.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024b. Model editing harms general abilities of large language models: Regularization to the rescue. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 16801–16819. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7969–7992. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831. PMLR.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024a. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791, Mexico City, Mexico. Association for Computational Linguistics.

Yucheng Shi, Qiaoyu Tan, Xuansheng Wu, Shaochen Zhong, Kaixiong Zhou, and Ninghao Liu. 2024b. Retrieval-enhanced knowledge editing in language models for multi-hop question answering.

Weihang Su, Yiran Hu, Anzhe Xie, Qingyao Ai, Zibing Que, Ning Zheng, Yun Liu, Weixing Shen, and Yiqun Liu. 2024a. Stard: A chinese statute retrieval dataset with real queries issued by non-professionals.

Weihang Su, Yichen Tang, Qingyao Ai, Changyue Wang, Zhijing Wu, and Yiqun Liu. 2024b. Mitigating entity-level hallucination in large language models. *arXiv preprint arXiv:2407.09417*.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024c. Unsupervised real-time hallucination detection based on the internal states of large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14379–14391, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Weihang Su, Changyue Wang, Anzhe Xie, Qingyao Ai, Yiran Hu, and Yiqun Liu. 2024d. Legalaid: A large language model for the chinese legal field. `https://github.com/oneal2000/LegalAID`.

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

Changyue Wang, Weihang Su, Yiran Hu, Qingyao Ai, Yueyue Wu, Cheng Luo, Yiqun Liu, Min Zhang, and Shaoping Ma. 2024a. Lekube: A knowledge update benchmark for legal domain. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 175–185.

Changyue Wang, Weihang Su, Hu Yiran, Qingyao Ai, Yueyue Wu, Cheng Luo, Yiqun Liu, Min Zhang, and Shaoping Ma. 2024b. Lekube: A legal knowledge update benchmark.

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. 2024c. Rat: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation. *Preprint*, arXiv:2403.05313.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian

Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *ArXiv preprint*, abs/2407.10671.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5823–5840, Toronto, Canada. Association for Computational Linguistics.

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? 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 Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15686–15702, Singapore. Association for Computational Linguistics.

A Pseudocode Description of EditCoT

The pseudocode description of EditCoT is shown in Algorithm 1.

B Detailed Comparison with the Verify-and-Edit Framework

Zhao et al. (2023) introduce a Verify-and-Edit framework that, on the surface, appears similar to our approach as it also involves editing the generated CoT. However, there are several key distinctions: (1) their verification process relies on the self-consistency, which works well for general QA tasks but is less applicable to knowledge editing since the model often shows high confidence in outdated information, leading to high consistency even when incorrect. We conducted tests on MQuAKE-CF-3k, focusing only on the triggering of editing and ignoring subsequent edit results. Using the self-consistency, edits were triggered in only 37% of cases on the knowledge editing dataset, while our method triggered edits in 95% of cases.; (2) their editing relies on the model's ability to learn from context, whereas in Table 5.4, we demonstrate that our proposed CoT editor significantly outperforms this self-editing approach in knowledge editing tasks; and (3) their method actually resembles Mello's (Zhong et al., 2023) strategy of decomposing tasks, whereas our approach modifies the entire CoT in each round of editing, allowing for more comprehensive adjustments.

C Further Discussion on Relationship Between EditCoT and RAG

This section provides a focused discussion on the conceptual and empirical distinctions between EditCoT and traditional RAG-based approaches, in response to the common concern that EditCoT may overlap significantly with existing retrieval-based methods.

C.1 Beyond Knowledge-Item Conflicts: Reasoning-Path Level Editing

The RAG method occasionally faces conflicts between the knowledge retrieved and the knowledge embedded within the LLM parameters. Techniques like contrastive or context-aware decoding (Li et al., 2023; Shi et al., 2024a) aim to enforce attention to the retrieved context. However, these approaches generally assume that the LLM merely ignores context, but overlook a deeper issue: the model may integrate the context while still generating a reasoning path that is logically inconsistent with it. For instance, we observe that when the context presents edited knowledge, the model's reasoning path focuses on clarifying and rectifying the knowledge within the context. In this scenario, the model's error does not stem from overlooking the context, and therefore contrastive decoding does not address the problem.

In contrast, EditCoT is designed to address precisely these reasoning-path conflicts. Even when the model attends to updated knowledge, its internal chain of thought (CoT) may still follow an outdated or incompatible reasoning structure. EditCoT proposes an iterative editing paradigm that identifies such inconsistencies and revises the CoT

Algorithm 1 EditCoT Framework

```
**Input:** Query $q$, Original Model $M$, Edited Model $M_{\text{edit}}$, Memory of Facts $K$, Maximum Iterations $N$, Mapping of old facts to new facts $F$
**Output:** Final Answer $A_{\text{final}}$
$A_{\text{old}} \leftarrow \text{GenerateAnswer}(M, q)$    ▷ Generate initial answer
$CoT_{\text{old}} \leftarrow \text{GenerateCoT}(M, q, A_{\text{old}})$    ▷ Generate initial chain-of-thought
for $i = 1$ to $N$ do
    for each step $s$ in $CoT_{\text{old}}$ do
        $F_{\text{new}} \leftarrow \text{RetrieveFacts}(s, K)$    ▷ Retrieve new fact from memory
        if $F$ exists then
            $F_{\text{new}} \leftarrow F[F_{\text{new}}]$    ▷ Map old fact to new fact if applicable
        end if
        if $F_{\text{new}} \neq \varnothing$ then
            break
        end if
    end for
    $is\_Conflict \leftarrow \text{Verify}(M, q, CoT_{\text{old}}, F_{\text{new}})$    ▷ Verify conflict between CoT and new fact
    if not $is\_Conflict$ then
        break
    end if
    $CoT_{\text{new}} \leftarrow \text{EditChain}(M_{\text{edit}}, q, CoT_{\text{old}}, F_{\text{new}})$    ▷ Edit chain-of-thought with new fact
    $CoT_{\text{old}} \leftarrow CoT_{\text{new}}$    ▷ Update old CoT with new one
end for
$A_{\text{final}} \leftarrow \text{GenerateAnswer}(M, CoT_{\text{new}})$    ▷ Generate final answer based on edited CoT
return $A_{\text{final}}$
```

accordingly, without requiring internal model access. This makes EditCoT applicable to both open- and closed-source models, such as GPT-4o.

We highlight several key differences between EditCoT and RAG-based contrastive decoding methods:

- **Conflict level:** EditCoT edits reasoning paths iteratively, handling subtle logical inconsistencies, while contrastive decoding typically handles direct knowledge-level conflicts through token-level or intermediate-layer hard enforcement. EditCoT can be seen as a soft method.
- **Model access:** EditCoT does not require internal model access, making it universally applicable even to closed-source large models like GPT-4o (see our experiments on GPT-4o, Table 3), whereas contrastive decoding approaches often need white-box access, limiting their broader usability.
- **Pipeline structure:** EditCoT includes a full ICE (In-Context Editing) pipeline—retrieval, conflict detection, and CoT editing—tailored specifically for knowledge editing. Traditional RAG approaches (including contrastive decoding methods) typically lack such integrated frameworks.

To empirically validate these differences, we conducted an additional experiment on the MQuAKE-CF-3k benchmark using Llama3-8B-Instruct. We employ CAD (Context-aware Decoding (Shi et al., 2024a), a contrastive decoding method) to enhance two baselines, utilizing CAD techniques when the context contains retrieved edit knowledge or updated reasoning paths. Given its features of subproblem decomposition and multi-hop retrieval process, and the absence of other optimizations, we regard Mello as a multi-hop RAG technique in this context. Results are summarized in Table 7.

While CAD does enhance Mello's performance, it continues to lag significantly behind EditCoT. These results empirically demonstrate that EditCoT more effectively mitigates reasoning-path conflicts than traditional RAG or contrastive decoding approaches. We interpret this as evidence that purely "hard" decoding strategies do not sufficiently address deeper conflicts in the reasoning process during knowledge editing. In contrast, EditCoT's holis-

Table 7: A comparison between RAG + CAD technology and EditCoT. We further explore the enhanced performance of EditCoT when utilizing CAD. MH-Accuracy represents the Multihop-Accuracy.

Method	MH-Accuracy (%)
Mello (multi-hop RAG)	10.0
Mello + CAD	13.3
EditCoT	35.4
EditCoT + CAD	39.5

tic approach, featuring iterative revisions, is inherently better suited for resolving such conflicts. Crucially, integrating EditCoT with CAD is not contradictory; rather, the combination of EditCoT's framework and contrastive decoding leads to additional performance improvements.

C.2 In-Context Editing and RAG

Current In-Context Editing (ICE) methods are typically grounded in a retrieval process, and thus can be considered variants of Retrieval-Augmented Generation (RAG). However, general RAG methods are initially developed for open-domain question answering and general knowledge-intensive tasks, and ICE has emerged as a specialized framework tailored to the demands of knowledge editing. These tasks often involve small-scale, precise, and repetitive updates to the model's underlying knowledge, which are not well-served by standard RAG pipelines.

Unlike RAG, which retrieves relevant information to supplement the model's parametric knowledge, ICE methods aim to directly guide the model in revising its reasoning in light of new or corrected knowledge. For example, some ICE variants reformulate edits as the answers to intermediate sub-questions (e.g., PokeMQA (Gu et al., 2024a)), while others construct structured representations for targeted editing (e.g., RAE (Shi et al., 2024b)). Compared to the standard RAG approach, these adjustments make the ICE method more suitable for knowledge editing scenarios. Building on the limitations identified in these methods, EditCoT further investigates and refines their underlying principles.

However, prior ICE methods do not discuss the specific differences or potential improvements that ICE offers over traditional RAG. To address these limitations, in our main experiments (Table 1), we extended our baseline comparisons to include not just "naïve RAG" but also two more advanced RAG methods, showing that applying conventional RAG to knowledge editing remains suboptimal without additional adjustments. We explore the reasons for this outcome in subsection 5.1.

D Details of Benchmarks

Table 8 shows the statistics of the three benchmarks, including the total number of edits and queries.

Table 8: Statistics on the number of edits and questions for the benchmarks

Dataset	Subset	Edits	Queries
MQuAKE-CF-3k	-	2785	3,000
DUNE	Scientific Reasoning	223	1508
	Arithmetic Reasoning	184	1065
	New Information	200	1000
LeKUBE	MCQ of the Legal Scenario	180	180
	T/F Questions of Change in Statute	180	642

E Details of Baselines

We focus on In-Context Editing methods and also compare EditCoT with several traditional model editing techniques:

Fine-tuning (FT): A full-parameter fine-tuning approach. Since we evaluate Instruct Models, we train the model on question-answer pairs constructed from each dataset's editing instances.

Naive RAG (Lewis et al., 2020): Given an external knowledge base and a retriever, the Retrieval-Augmented Generation (RAG) framework injects relevant knowledge into the model by concatenating retrieved documents into the context, effectively augmenting the language model's ability to reason and reduce hallucinations (Su et al., 2024c; Chen et al., 2024; Su et al., 2024b,d). Following the settings from the original LeKUBE and DUNE papers, we use BM25(Robertson et al., 2009) as the retriever, a retrieval method based on lexical matching, and concatenate 3 and 1 retrieved documents, respectively. For the evaluation of MQuAKE-CF-3k, since the original paper does not test RAG, we still use BM25 as the retriever, setting it to concatenate 5 retrieved documents.

RAT(Wang et al., 2024c): A RAG method that refines the model-generated reasoning steps through multiple rounds of query rewriting and retrieval. We utilize the official prompt template and ultimately guided the model to derive the final answer based on the original question and the final reasoning steps. All other settings are identical to those in Naive RAG.

FLARE(Jiang et al., 2023): A dynamic RAG that determines whether to perform retrieval by assessing the uncertainty of the generated sentences during the generation process. We set the hallucination detection threshold at 0.12. All other settings remain the same as in Naive RAG.

KN(Dai et al., 2022): This method updates knowledge by identifying and editing "knowledge neurons" in the model. In our experiments, we set the prompt number n for neuron identification to 10, the knowledge attribution threshold to 0.2, and the probability of retaining shared neurons to 0.4.

ROME(Meng et al., 2022): A Rank-One Model Editing method that directly modifies key-value pairs in the FFN layers. For our experiments, we uniformly modify the fifth layer and calculate the loss at the final layer of the model. The weight decay is set to 1×10^{-3}.

Both KN and ROME methods are implemented using the EasyEdit(Wang et al., 2023a) library, which integrates several model editing techniques for ease of use. It is important to note that both methods rely on knowledge triples for editing. In the *Scientific Reasoning* and *New Information* subsets of DUNE, the edits consist of longer natural language sequences. We use GPT-4o to extract the relevant knowledge triples.

Mello(Zhong et al., 2023): Mello performs in-context editing by decomposing the problem into sub-questions and utilizing retrieval. We follow the official setting with a maximum of 4 retrieval rounds and adapt the prompts to Instruct Models. In English datasets, we use Contriever(Izacard et al., 2022) as the retriever, following the settings from the original paper. For the Chinese LeKUBE dataset, we translate the prompts into Chinese and use BM25 as the retriever.

PokeMQA(Gu et al., 2024a): Built on top of Mello, PokeMQA adds entity extraction and scope determination to refine the question understanding. We follow the official setting with a maximum of 5 rounds and use their pre-trained Scope-Detector. For DUNE and LeKUBE, where no pre-extracted entities are available, we instruct the model to extract entities in a similar format. Since the official Scope-Detector only supports English, we translate the input sentences for LeKUBE tasks into English before applying the detector.

During the GPT-4o evaluations, we observed difficulties in strictly following the prompt format. To mitigate this issue, we add the system prompt: "Follow the examples below."

Additionally, for Mello, PokeMQA and EditCoT, since LeKUBE evaluates changes in legal provisions, and other baselines can utilize both old and new laws, we concatenate the pre- and post-update legal texts and provide them to the model after retrieval for fair comparison.

RAE(Shi et al., 2024b): RAE constructs knowledge graphs for retrieval and leverages the model to assist in retrieving and pruning the graphs. However, for DUNE and LeKUBE, knowledge graphs similar to those in Wikipedia cannot be constructed, making RAE less generalizable to these tasks.

Additionally, the LLMs we used are implemented by Huggingface Transformers library (Wolf et al., 2020).

F Details of Efficiency Experiments

All efficiency experiments are conducted on a single NVIDIA A100 (40G) GPU.

RAE's official implementation uses Wikipedia API calls; hence, we exclude network latency from its reported inference time (including this would place RAE around 46 seconds on the x-axis). Despite this, RAE has a notably longer inference time due to multiple forward propagations for knowledge graph retrieval and pruning, particularly when selecting the next relation, which requires feeding all candidates into the model respectively.

The efficiency of EditCoT is attributed to fewer required iterations: while PokeMQA averages 4.33 iterations, EditCoT needs only 3.65, indicating fewer long-text generation steps. Here a full retrieval-generation cycle is defined as one iteration.

G Performance with different max iterations

We explore how varying the maximum number of iterations affects their performance. Figure 6 shows the results. PokeMQA and Mello require more iterations to reach optimal performance, with relatively low accuracy in early iterations. On the other hand, EditCoT outputs a complete chain-of-thought in each iteration, achieving decent performance after the first iteration.

H Investigating a SFT Baseline with HotpotQA Knowledge

To ensure the fairness of our comparison and confirm that any improvements in EditCoT are not due

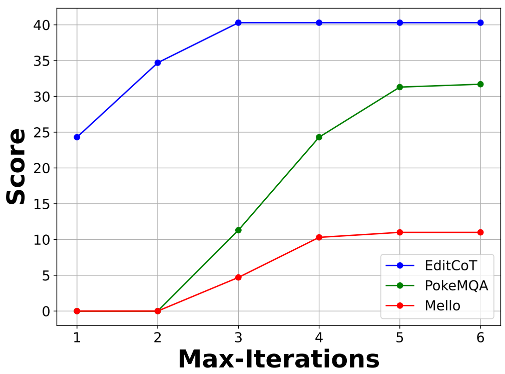


Figure 6: Performance with different max iterations. A full retrieval-generation cycle is defined as one iteration.

to knowledge leakage from the CoT editor's training process, we employ Supervised Fine-Tuning (SFT) on the Llama3-8B-Instruct, using it to test two baselines. The fine-tuning dataset is constructed using the correct reasoning paths sourced from the CoT Editor's training dataset, which is derived from HotpotQA. These reasoning paths are formatted into a question-answering (QA) structure suitable for SFT. The fine-tuned model (hereinafter SFT-Llama3-8B-Instruct) possesses a level of knowledge on the HotpotQA domain comparable to that accessible by EditCoT's CoT Editor. Subsequently, both the SFT-Llama3-8B-Instruct model and the original Llama3-8B-Instruct model are evaluated on the **MQuAKE-CF-3k** benchmark. The comparative results are presented in Table 9.

Table 9: Performance comparison on MQuAKE-CF-3k between the original Llama3-8B-Instruct model and its counterpart fine-tuned (SFT) with HotpotQA reasoning paths.

Model Configuration	Mello	PokeMQA
SFT Llama3-8B-Instruct	12.6	24.8
Original Llama3-8B-Instruct	10.0	26.1

We observe that the variations among the inference models lead to only minor differences, which do not translate into a significant advantage or leakage for the newly trained model, further supporting the fairness of our overall design.

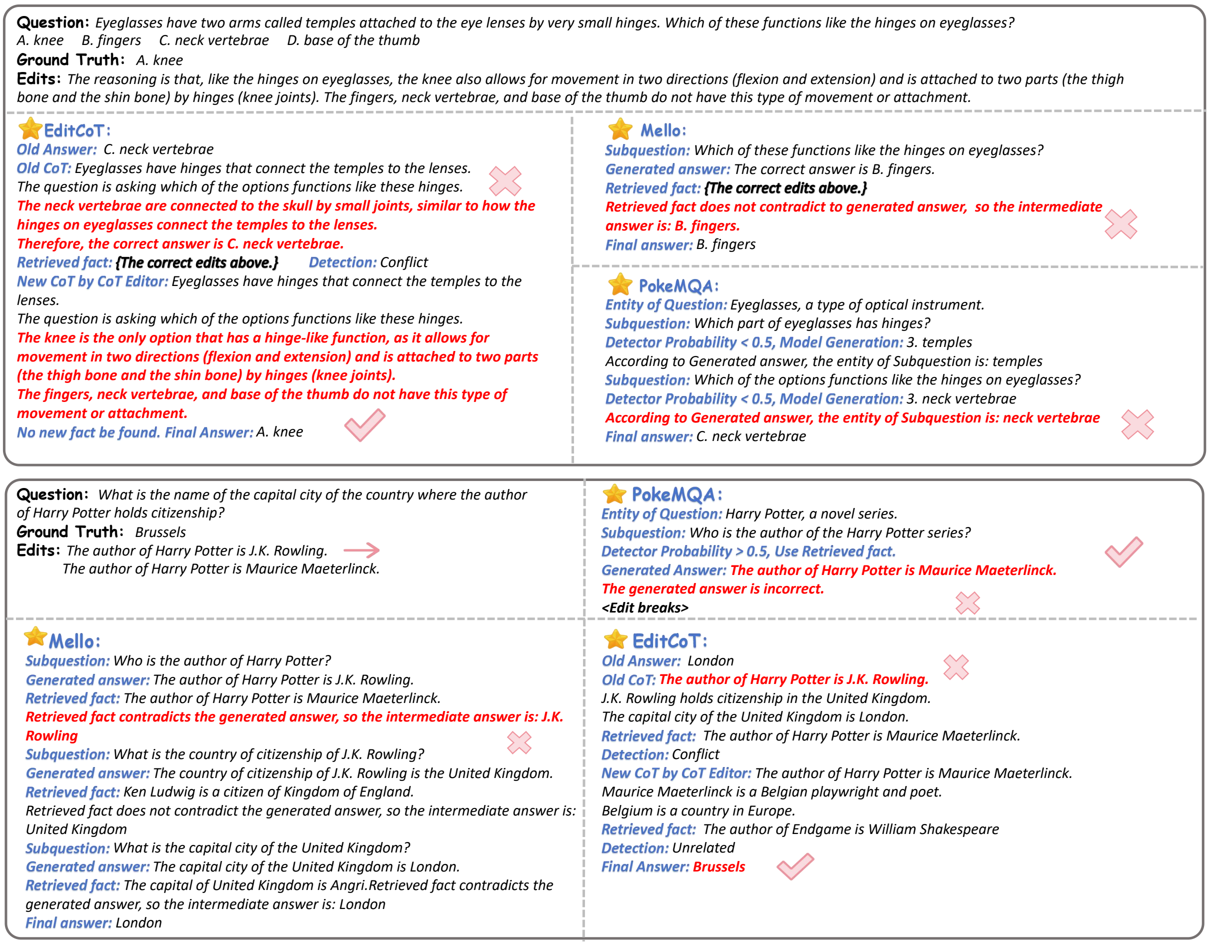
I Case Studies

In Figure 7, we compare EditCoT with Mello and PokeMQA on two questions. The first question from the Scientific Reasoning subset of DUNE shows that EditCoT initially generates an incorrect answer and CoT. However, after retrieval and conflict detection, it modifies the CoT to arrive at the correct solution. In contrast, Mello retrieves a relevant edit but fails to detect any conflict with the initial answer, missing the opportunity for correction. This demonstrates the strength of EditCoT's conflict detection mechanism, which evaluates the entire reasoning chain against retrieved edits. PokeMQA struggles with both sub-questions, producing a repetitive decomposition and failing to trigger an edit.

The second question from MQuAKE-CF-3k again illustrates EditCoT's ability to transition from an incorrect inference to the correct answer through CoT editing. Although both PokeMQA and Mello successfully retrieve and detect the relevant edit (changing the author of Harry Potter to Maurice Maeterlinck), they do not apply it. This indicates that models designed with safety in mind may resist incorporating external knowledge in favor of their internal knowledge. EditCoT addresses this issue by decoupling conflict detection from the editing process and training a model specifically for CoT editing, leading to more effective reasoning updates.

Additionally, Table 10 and Table 11 illustrate two failure cases of EditCoT, offering a more comprehensive perspective on its functionality. The error in Table 10 arises from issues in both the retriever and the conflict detection module. Specifically, the editing process should have correctly terminated after the first edit (i.e., "CoT-2 by CoT Editor"). However, due to suboptimal retriever performance, unrelated examples were retrieved, and conflict detection mistakenly flagged a conflict, leading EditCoT to perform an unnecessary additional edit, which resulted in an incorrect outcome. To maintain fairness in experimental comparisons, EditCoT shares the same retriever with other ICE methods (e.g., Mello), and thus this module was not specifically optimized. It is worth noting that such errors are rare (approximately 1-2 cases out of 200 manually inspected examples), as it is uncommon for both the retriever and conflict detection modules to fail simultaneously. However, this suggests potential future improvements for conflict detection.

In contrast, the error in Table 11 is entirely due to the original LLM's mistake. The question concerns the origins of certain sports, but the original CoT generated by the model instead provides information about club locations. As a result, the retrieval



Question: *Eyeglasses have two arms called temples attached to the eye lenses by very small hinges. Which of these functions like the hinges on eyeglasses?*
A. knee B. fingers C. neck vertebrae D. base of the thumb
Ground Truth: *A. knee*
Edits: *The reasoning is that, like the hinges on eyeglasses, the knee also allows for movement in two directions (flexion and extension) and is attached to two parts (the thigh bone and the shin bone) by hinges (knee joints). The fingers, neck vertebrae, and base of the thumb do not have this type of movement or attachment.*

EditCoT:
Old Answer: *C. neck vertebrae*
Old CoT: *Eyeglasses have hinges that connect the temples to the lenses.*
The question is asking which of the options functions like these hinges. ✗
The neck vertebrae are connected to the skull by small joints, similar to how the hinges on eyeglasses connect the temples to the lenses.
Therefore, the correct answer is C. neck vertebrae.
Retrieved fact: ***{The correct edits above.}*** *Detection:* *Conflict*
New CoT by CoT Editor: *Eyeglasses have hinges that connect the temples to the lenses.*
The question is asking which of the options functions like these hinges.
The knee is the only option that has a hinge-like function, as it allows for movement in two directions (flexion and extension) and is attached to two parts (the thigh bone and the shin bone) by hinges (knee joints).
The fingers, neck vertebrae, and base of the thumb do not have this type of movement or attachment.
No new fact be found. Final Answer: *A. knee* ✓

Mello:
Subquestion: *Which of these functions like the hinges on eyeglasses?*
Generated answer: *The correct answer is B. fingers.*
Retrieved fact: ***{The correct edits above.}***
Retrieved fact does not contradict to generated answer, so the intermediate answer is: B. fingers. ✗
Final answer: *B. fingers*

PokeMQA:
Entity of Question: *Eyeglasses, a type of optical instrument.*
Subquestion: *Which part of eyeglasses has hinges?*
Detector Probability < 0.5, Model Generation: *3. temples*
According to Generated answer, the entity of Subquestion is: temples
Subquestion: *Which of the options functions like the hinges on eyeglasses?*
Detector Probability < 0.5, Model Generation: *3. neck vertebrae*
According to Generated answer, the entity of Subquestion is: neck vertebrae ✗
Final answer: *C. neck vertebrae*

Question: *What is the name of the capital city of the country where the author of Harry Potter holds citizenship?*
Ground Truth: *Brussels*
Edits: *The author of Harry Potter is J.K. Rowling.* →
The author of Harry Potter is Maurice Maeterlinck.

PokeMQA:
Entity of Question: *Harry Potter, a novel series.*
Subquestion: *Who is the author of the Harry Potter series?*
Detector Probability > 0.5, Use Retrieved fact. ✓
Generated Answer: ***The author of Harry Potter is Maurice Maeterlinck.***
The generated answer is incorrect.
<Edit breaks> ✗

Mello:
Subquestion: *Who is the author of Harry Potter?*
Generated answer: *The author of Harry Potter is J.K. Rowling.*
Retrieved fact: *The author of Harry Potter is Maurice Maeterlinck.*
Retrieved fact contradicts the generated answer, so the intermediate answer is: J.K. Rowling ✗
Subquestion: *What is the country of citizenship of J.K. Rowling?*
Generated answer: *The country of citizenship of J.K. Rowling is the United Kingdom.*
Retrieved fact: *Ken Ludwig is a citizen of Kingdom of England.*
Retrieved fact does not contradict the generated answer, so the intermediate answer is: United Kingdom
Subquestion: *What is the capital city of the United Kingdom?*
Generated answer: *The capital city of the United Kingdom is London.*
Retrieved fact: *The capital of United Kingdom is Angri.Retrieved fact contradicts the generated answer, so the intermediate answer is: London*
Final answer: *London*

EditCoT:
Old Answer: *London* ✗
Old CoT: ***The author of Harry Potter is J.K. Rowling.***
J.K. Rowling holds citizenship in the United Kingdom.
The capital city of the United Kingdom is London.
Retrieved fact: *The author of Harry Potter is Maurice Maeterlinck.*
Detection: *Conflict*
New CoT by CoT Editor: *The author of Harry Potter is Maurice Maeterlinck.*
Maurice Maeterlinck is a Belgian playwright and poet.
Belgium is a country in Europe.
Retrieved fact: *The author of Endgame is William Shakespeare*
Detection: *Unrelated*
Final Answer: ***Brussels*** ✓

Figure 7: A face-to-face comparison of our proposed EditCoT method with other baseline approaches. The first example comes from the Scientific Reasoning subset of DUNE. The second example comes from MQuAKE-CF-3k. The model here is Meta-Llama-3-8B-Instruct.

process produces no results, preventing EditCoT from proceeding as intended. It is important to emphasize that EditCoT is designed to enable reasoning with updated knowledge after editing, rather than to enhance the original model's reasoning capabilities or overall performance. From our manual inspection, such errors are more prevalent and should be attributed to the reasoning limitations of the original model rather than any shortcomings of the CoT Editor itself.

J Prompt Templates

Since the models we are working with are all Instruct Models, the following prompt template uses "system", "user", and "assistant" to represent different roles.

J.1 Prompt Templates of Dataset Construction

First, we use RAG to generate CoT and the answer. The prompt template is as follows:

Table 10: A failure case example for EditCoT, derived from the results of Llama-3-8B-Instruct on MQuAKE-CF-3k. We highlight the first occurrence of the error in red. The Ground Truth Reasoning Path is the reasoning path provided by the dataset that the edited model is expected to output.

Question: What sport is associated with the employer of John Farrell?	
Related Edits: John Farrell is employed by BOSTON RED SOX → FC Groningen **Ground Truth Reasoning Path:** John Farrell is employed by FC Groningen FC Groningen is associated with the sport of association football	
CoT-1 (The original CoT)	John Farrell is the manager of the Boston Red Sox. The Boston Red Sox is a team in the sport of Baseball.
Retrieval	**Query:** John Farrell is the manager of the Boston Red Sox. **Result:** John Farrell is employed by FC Groningen **Detection:** Conflict
CoT-2 by CoT Editor	John Farrell is the manager of the FC Groningen. FC Groningen is associated with the sport of association football.
Retrieval	**Query:** FC Groningen is associated with the sport of association football. **Result:** AGOVV Apeldoorn is associated with the sport of baseball **Detection:** Conflict
CoT-3 by CoT Editor	John Farrell is the manager of AGOVV Apeldoorn. The AGOVV Apeldoorn is a team in the sport of baseball.

Table 11: A failure case example for EditCoT, derived from the results of Llama-3-8B-Instruct on MQuAKE-CF-3k. We highlight the first occurrence of the error in red. The Ground Truth Reasoning Path is the reasoning path provided by the dataset that the edited model is expected to output.

Question: Which country or location is the origin of the sport associated with NK Nafta Lendava?	
Related Edits: association football was created in England → Hong Kong **Ground Truth Reasoning Path:** NK Nafta Lendava is associated with the sport of association football association football was created in Hong Kong	
CoT-1 (The original CoT)	NK Nafta Lendava is a football club. The sport associated with NK Nafta Lendava is football (soccer). The country where NK Nafta Lendava is located is Slovenia.
Retrieval	No results in retrieval.
CoT-2 by CoT Editor	NK Nafta Lendava is a football club. The sport associated with NK Nafta Lendava is football (soccer). The country where NK Nafta Lendava is located is Slovenia.

Dataset Construction - RAG Prompt

User: {*Top-5 relevant paragraphs from Wikipedia*}

Instruction: You have access to background information that may assist in answering the question. Focus on reasoning through the problem step by step, keeping the explanation concise. Use the token [STEP] to start each thought step and the token [ANSWER] to indicate the final answer. Avoid mentioning or referencing the background information directly in your reasoning.

Template:
[STEP] ...
[STEP] ...
...
[ANSWER] ...

Question: {*A question from HotpotQA*}

Second, we conduct prefix-guided CoT generation. The prompt template is as follows:

Dataset Construction - Prompt of prefix-guided CoT generation

User: Please answer the following question using a chain-of-thought. Use the token [STEP] to start each thought step, and the token [ANSWER] to indicate the final answer. Keep each step brief and to the point and keep the final answer concise.

Template:
[STEP] ...
[STEP] ...
...
[ANSWER] ...

Question: {*A question from HotpotQA*}
Assistant: [STEP] {*The first (k-1) steps of the CoT from RAG*}
{...}

Then, we detect conflict in the final answers:

Dataset Construction - Prompt of Answer Conflict Detection

User: For the given question Q, and the two provided answers (Answer 1 and Answer 2), determine if the answers are semantically consistent. Choose the most appropriate option from the following:

A. Yes, they convey similar ideas or information.
B. No, they are different or contradictory.

Question Q: {*A question from HotpotQA*}
Answer 1: {*The answer generated by RAG*}
Answer 2: {*The answer generated by prefix-guided CoT generation*}

Your choice:

And detect if there is a conflict at the first diverging step between the two CoTs:

Dataset Construction - Prompt of CoT Step Conflict Detection

User: Evaluate the relationship between the following two sentences based on their factual content. Choose the most appropriate option from the following:

A. The two sentences contain conflicting knowledge.
B. The two sentences support or complement each other.
C. The two sentences are unrelated (no conflict, but no connection).

Sentence 1: {*CoT Step k*}
Sentence 2: {*CoT Step* k^*}

Your choice:

Finally, we let the model rewrite the key step:

Dataset Construction - Rewriting Prompt

User: Rewrite the following sentence in the style of a Wikipedia data, using formal and objective language. Only express the factual knowledge present in the sentence, without adding any extra information or inventing details. Your output must be one or more sentences, only needs to include direct results without extra words.

Input:

Sentence: {*A CoT step that needs to rewrite*}

Output:

J.2 Prompt Templates of EditCoT

The prompt template for CoT editor is as follows. Note that the template is the same during both the training and inference phases.

EditCoT Framework - Prompt for CoT editor

System: You can edit the following chains of thought based on the new knowledge.
User: Question: {*A user question*}
Old Thoughts: {*A old CoT*}
New knowledge: {*A sentence containing new knowledge*}
Please give me the new chain-of-thought based on the new knowledge.
Assistant: New Thoughts:

The prompt templates below are for the original LLM. First, we instruct it to generate an answer directly:

EditCoT Framework - Prompt of Answering directly

User: Question: What is the capital of the country where Plainfield Town Hall is located?
Assistant: Answer: Washington, D.C.

User: Question: In which country is the company that created Nissan 200SX located?
Assistant: Answer: Japan

User: Question: Which continent is the country where the director of "My House Husband: Ikaw Na!" was educated located in?
Assistant: Answer: Asia

User: Question: Who is the spouse of the US president?
Assistant: Answer: Jill Biden

User: Question: Who has ownership of the developer of the Chevrolet Corvette (C4)?
Assistant: Answer: General Motors

User: Question: {*A user question*}
Assistant: Answer:

The prompt template of get an original CoT from a question and an answer is as follows:

EditCoT Framework - Prompt of Getting the original CoT

User: Question: What is the capital of the country where Plainfield Town Hall is located?
Answer: Washington, D.C.
Please give the chain of thought based on the question and answer pairs above.
Assistant: Thoughts: Plainfield Town Hall is located in the country of the United States of America.
The capital of United States is Washington, D.C.

User: Question: In which country is the company that created Nissan 200SX located?
Answer: Japan
Please give the chain of thought based on the question and answer pairs above.
Assistant: Thoughts: Nissan 200SX was created by Nissan.
Nissan is located in the country of Japan.

User: Question: Which continent is the country where the director of "My House Husband: Ikaw Na!" was educated located in?
Answer: Asia

Please give the chain of thought based on the question and answer pairs above.
Assistant: Thoughts: 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 of Asia.

User: Question: Who is the spouse of the US president?
Answer: Jill Biden
Please give the chain of thought based on the question and answer pairs above.
Assistant: Thoughts: The US president is Joe Biden.
The spouse of Joe Biden is Jill Biden.

User: Question: Who has ownership of the developer of the Chevrolet Corvette (C4)?
Answer: General Motors
Please give the chain of thought based on the question and answer pairs above.
Assistant: Thoughts: The developer of Chevrolet Corvette (C4) is Chevrolet.
Chevrolet is owned by General Motors.

User: Question: {*A user question*}
Answer: {*An original answer*}
Please give the chain of thought based on the question and answer pairs above.
Assistant: Thoughts:

The prompt template of conflict detection is as follows. We use 5-shot examples and apply a scenario similar to CoT editor.

EditCoT Framework - Prompt of Conflict Detection

System: You can edit the following chains of thought based on the new knowledge.
User: Question: Which country is the author of M̈oneyballä citizen of?
Old Thoughts: The author of Moneyball is Michael Lewis.
Michael Lewis is a citizen of United States of America.
New knowledge: The author of Our Mutual Friend is Charles Dickens
Please give me the new chain-of-thought based on the new knowledge.
Assistant: Relevance of knowledge and chain-of-thought: Unrelated.

User: Question: What language does the Shadow Home Secretary speak, write, or sign?
Old Thoughts: The Shadow Home Secretary is Diane Abbott.
Diane Abbott speaks the language of English.
New knowledge: The Shadow Home Secretary is Diane Abbott
Please give me the new chain-of-thought based on the new knowledge.
Assistant: Relevance of knowledge and chain-of-thought: Support.

User: Question: What is the notable work of the author of T̈his Side of Paradise?̈
Old Thoughts: The author of This Side of Paradise is F. Scott Fitzgerald.
F. Scott Fitzgerald is famous for The Great Gatsby.
New knowledge: The author of This Side of Paradise is Geoffrey Chaucer
Please give me the new chain-of-thought based on the new knowledge.
Assistant: Relevance of knowledge and chain-of-thought: Contradict.
New Thoughts: The author of This Side of Paradise is Geoffrey Chaucer.
Geoffrey Chaucer is famous for Shaman King.

User: Question: Who is the head of government of the country where Roy Ascott holds citizenship?
Old Thoughts: Roy Ascott is a citizen of United Kingdom.
The name of the current head of the United Kingdom government is Boris Johnson.
New knowledge: Roy Ascott is a citizen of United States of America
Please give me the new chain-of-thought based on the new knowledge.
Assistant: Relevance of knowledge and

chain-of-thought: Contradict.
New Thoughts: Roy Ascott is a citizen of United States of America.
The name of the current head of the United States of America government is Ole Johan Vierdal.

User: Question: Who is the spouse of the performer of the song D̈on't Be Cruel?̈
Old Thoughts: Don't Be Cruel was performed by Elvis Presley.
Elvis Presley is married to Priscilla Presley.
New knowledge: Don't Be Cruel was performed by Elvis Presley
Please give me the new chain-of-thought based on the new knowledge.
Assistant: Relevance of knowledge and chain-of-thought: Support.

User: Question: {*A user question*}
Old Thoughts: {*An old CoT*}
New knowledge: {*A sentence containing new knowledge*}
Please give me the new chain-of-thought based on the new knowledge.
Assistant: Relevance of knowledge and chain-of-thought:

The prompt of getting final answer is as follows. We instruct the original LLM to pay more attention to the new CoT.

EditCoT Framework - Prompt of Getting Final Answer

System: Suppose the world knowledge has been updated. Answer the question based on the [New Thoughts] rather than your own knowledge. Don't analyze whether the reasoning of [New Thoughts] is correct. Output the most concise answer directly.
User: Question: In which location are the headquarters of the developer of Android Jelly Bean situated?
[New Thoughts]: Android Jelly Bean was developed by Google.
The headquarters of Google is located in the city of Googleplex.
Assistant: Answer from [New Thoughts]: Googleplex

User: Question: Who is the head of government in the city where Husky Energy's headquarters is located?
[New Thoughts]: The headquarters of Husky Energy is located in the city of Calgary.
The name of the current head of the Calgary government is Jyoti Gondek.
Assistant: Answer from [New Thoughts]: Jyoti Gondek

User: Question: Who is the head of state of the country of citizenship of Theodoros Angelopoulos?
[New Thoughts]: Theodoros Angelopoulos is a citizen of Greece.
The name of the current head of state in Greece is Aikaterini Sakellaropoulou.
Assistant: Answer from [New Thoughts]: Aikaterini Sakellaropoulou

User: Question: Who is the head of government in the location where Vulcan Inc. is headquartered?
[New Thoughts]: The headquarters of Vulcan Inc. is located in the city of Seattle.
The name of the current head of the Seattle government is Bruce Harrell.
Assistant: Answer from [New Thoughts]: Bruce Harrell

User: Question: Who is the head of government of the country where Greg Combet holds citizenship?
[New Thoughts]: Greg Combet is a citizen of Australia.
The name of the current head of the Australia government is Anthony Albanese.
Assistant: Answer from [New Thoughts]: Anthony Albanese

User: Question: {*A question*}
[New Thoughts]: {*A CoT*}
Assistant: Answer from [New Thoughts]:

K Licensing

Qwen2.5-14B-Instruct is released under the Apache License 2.0. Meta-Llama-3-8B-Instruct is released under the META LLAMA 3 COMMUNITY LICENSE. KN, ROME, FT, FLARE are released under the MIT license. Contriever is released under the CC BY-SA 4.0 License.

The datasets MQuAKE, LeKUBE, and DUNE are released under the MIT license. This paper's research objective is academic exploration, which aligns with the terms of this license.