

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 updating¹.

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., 2024) 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

¹Code 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 re-training 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 para-

metric 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., 2024), 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

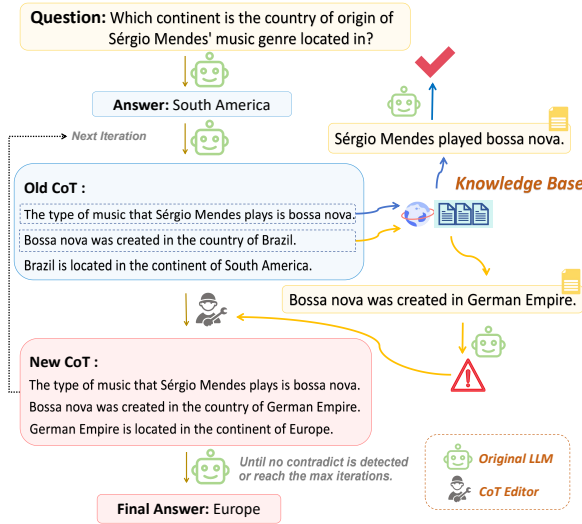


Figure 1: EditCoT updates LLMs by iteratively editing the CoT to integrate new knowledge while preserving its reasoning abilities. It generates an initial CoT, detects conflicts, and revises the CoT iteratively.

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 new, more accurate, or updated knowledge.

Step 3: Edit the chain-of-thought Once a knowledge conflict is detected, we input the question, new knowledge, and the old CoT into a CoT editor. This editor generates a new CoT based on the input. Then, our iterative process returns to Step 2, where this new CoT is treated as the old CoT for retrieval and conflict detection. The process continues until no conflicts are found, or the maximum number of iterations 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.

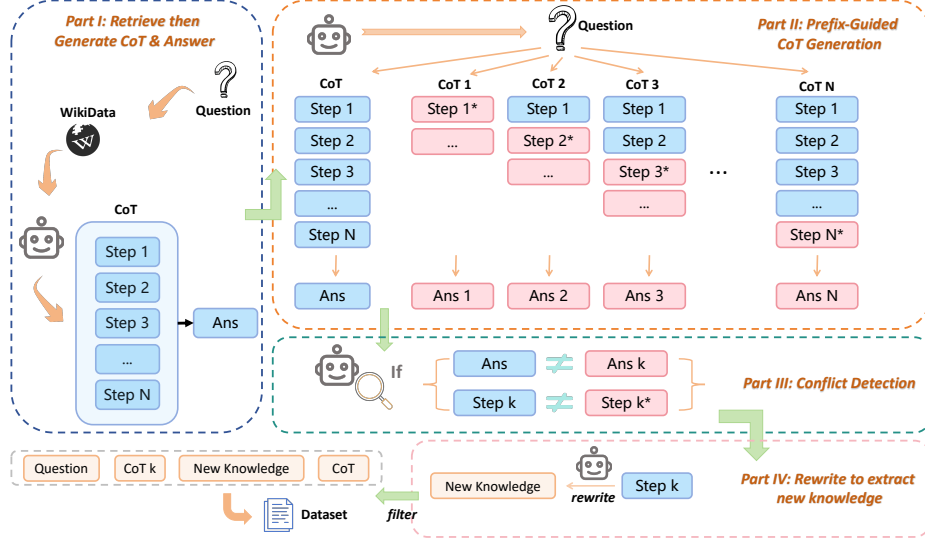


Figure 2: Diagram of the dataset construction: Our data source consists solely of questions from HotpotQA, without requiring human-annotated answers or related articles. In Part II, the blue sections represent the prefix used to guide generation, which have been generated in Part I, while red sections 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 k th step ($k = 1, 2, \dots, N$), 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:

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



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 H.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. The evaluation metric is *Multihop-Accuracy*, as in the original paper: each instance contains 3 questions, and if any one 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 C.

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., 2024), 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., 2024). 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 D](#).

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 H.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 G](#).

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’s success may depend more on the model’s internal reasoning abilities, whereas 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 we evaluate spanning multi-hop 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-3-8B-Instruct, and simpler methods like Naive RAG excel in narrow settings. However, these methods perform poorly on other datasets and 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 best or second-best

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	<u>35.4</u>
	Arithmetic	74.1	80.4	80.5	<u>84.0</u>	70.0	40.1	73.7	83.8	-	90.7
	DUNE Scientific	11.1	82.1	81.4	81.6	<u>84.3</u>	82.1	77.1	61.7	-	85.0
	New Info	30.3	66.6	69.1	89.2	77.0	81.6	<u>89.8</u>	58.7	-	91.3
Qwen	MQuAKE-CF-3k	9.3	3.2	0.0	10.0	3.3	3.7	5.8	5.3	<u>26.9</u>	34.2
	Arithmetic	82.8	86.9	86.9	<u>89.7</u>	82.6	23.4	61.0	35.2	-	97.3
	DUNE Scientific	13.8	75.3	75.5	70.7	<u>86.0</u>	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	-	<u>93.8</u>

Table 2: Applicability to vertical domains: LeKUBE is a Chinese Legal Knowledge Editing Benchmark. The model used is Qwen2.5-14B-Instruct.

Subsets of LeKUBE	Fine-tuning	Naive RAG	KN	ROME	Mello	PokeMQA	EditCoT(Ours)
MCQ of the Legal Scenario	42.2	<u>47.2</u>	38.9	38.3	6.7	0.0	58.3
T/F Questions of Change in Statute	65.9	<u>69.2</u>	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	<u>45.0</u>

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

performance across most datasets and models, although it is 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 5.4 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 old knowledge contained within its parameters.

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 is a crucial metric in knowledge editing, assessing a method’s ability to update specific knowledge without impairing untargated information (Mitchell et al., 2022). We evaluate lo-

cality 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 base 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,

²We 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.

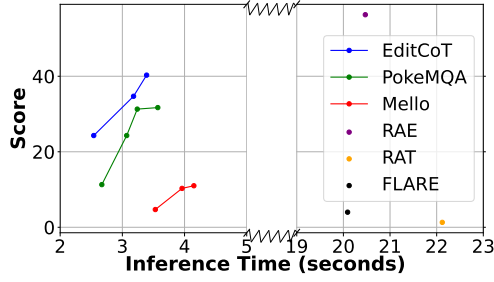


Figure 4: Inference time vs. performance: A method positioned in the top-left corner indicates shorter inference time and higher performance. Due to the extensive horizontal span, the figure is divided in the middle.

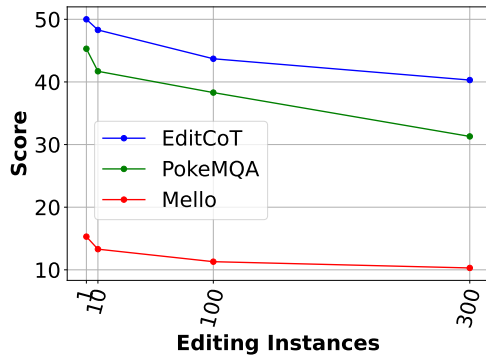


Figure 5: Performance with different batch sizes on MQuAKE-CF. The metrics here is multi-hop accuracy.

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.

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 E](#). 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 F](#). The average inference time for RAE and two advanced RAG methods exceeds 20 seconds, making their application in real-world tasks challenging.

Table 5: 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

5.4 Ablation Study

We conduct ablation studies using a dataset of 300 examples sampled from MQuAKE-CF-3k, with the model Qwen2.5-14B-Instruct.

Impact of Components We evaluate two components of our method. First, removing conflict detection before editing the CoT results in a 18.1% performance drop (33.0 vs. 40.3). Second, using the original model as the CoT editor without specialized training leads to an even more significant performance degradation (26.7 vs. 40.3). 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.

Impact of Training Configurations [Table 5](#) illustrates the effect of training dataset sizes and epochs on the CoT editor. 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. Additionally, the iterative nature of the CoT refinement process may add to the computational overhead.

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 Al-lonsius, 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-Yearly, 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, Shao-liang Nie, Sharan Narang, Sharath Rapparthi, 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, Wenxin Fu, Whit-

728	ney 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, Dingkan 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 Damla, 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 Tsim-poukelli, 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,	
729	Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev,	792
730	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 Ratan-chandani, 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, Vitor 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 .	793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829
731	Hengrui Gu, Kaixiong Zhou, Xiaotian Han, Ninghao Liu, Ruobing Wang, and Xin Wang. 2024a. PokeMQA: Programmable knowledge editing for multi-hop question answering . In <i>Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 8069–8083, Bangkok, Thailand. Association for Computational Linguistics.	830 831 832 833 834 835 836 837
732	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 <i>Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024</i> , pages 16801–16819. Association for Computational Linguistics.	838 839 840 841 842 843 844 845 846
733	Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning . <i>Trans. Mach. Learn. Res.</i> , 2022.	847 848 849 850 851
734	Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun,	852

853	Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie	Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-	912
854	Callan, and Graham Neubig. 2023. Active retrieval	Lopes, Jonathan Gordon, Morgan Grafstein, Scott	913
855	augmented generation . In <i>Proceedings of the 2023</i>	Gray, Ryan Greene, Joshua Gross, Shixiang Shane	914
856	<i>Conference on Empirical Methods in Natural Lan-</i>	Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris,	915
857	<i>guage Processing, EMNLP 2023, Singapore, Decem-</i>	Yuchen He, Mike Heaton, Johannes Heidecke, Chris	916
858	<i>ber 6-10, 2023</i> , pages 7969–7992. Association for	Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele,	917
859	Computational Linguistics.	Brandon Houghton, Kenny Hsu, Shengli Hu, Xin	918
860	Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yu-	Hu, Joost Huizinga, Shantanu Jain, Shawn Jain,	919
861	taka Matsuo, and Yusuke Iwasawa. 2022. Large lan-	Joanne Jang, Angela Jiang, Roger Jiang, Haozhun	920
862	guage models are zero-shot reasoners . In <i>Advances</i>	Jin, Denny Jin, Shino Jomoto, Billie Jonn, Hee-	921
863	<i>in Neural Information Processing Systems 35: An-</i>	woo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Ka-	922
864	<i>annual Conference on Neural Information Processing</i>	mali, Ingmar Kanitscheider, Nitish Shirish Keskar,	923
865	<i>Systems 2022, NeurIPS 2022, New Orleans, LA, USA,</i>	Tabarak Khan, Logan Kilpatrick, Jong Wook Kim,	924
866	<i>November 28 - December 9, 2022</i> .	Christina Kim, Yongjik Kim, Jan Hendrik Kirch-	925
867	Patrick S. H. Lewis, Ethan Perez, Aleksandra Pik-	ner, Jamie Kiros, Matt Knight, Daniel Kokotajlo,	926
868	tus, Fabio Petroni, Vladimir Karpukhin, Naman	Łukasz Kondraciuk, Andrew Kondrich, Aris Kon-	927
869	Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih,	stantinidis, Kyle Kosic, Gretchen Krueger, Vishal	928
870	Tim Rocktäschel, Sebastian Riedel, and Douwe	Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan	929
871	Kiela. 2020. Retrieval-augmented generation for	Leike, Jade Leung, Daniel Levy, Chak Ming Li,	930
872	knowledge-intensive NLP tasks . In <i>Advances in Neu-</i>	Rachel Lim, Molly Lin, Stephanie Lin, Mateusz	931
873	<i>ral Information Processing Systems 33: Annual Con-</i>	Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue,	932
874	<i>ference on Neural Information Processing Systems</i>	Anna Makanju, Kim Malfacini, Sam Manning, Todor	933
875	<i>2020, NeurIPS 2020, December 6-12, 2020, virtual</i> .	Markov, Yaniv Markovski, Bianca Martin, Katie	934
876	Kevin Meng, David Bau, Alex Andonian, and Yonatan	Mayer, Andrew Mayne, Bob McGrew, Scott Mayer	935
877	Belinkov. 2022. Locating and editing factual associ-	McKinney, Christine McLeavey, Paul McMillan,	936
878	ations in GPT . In <i>Advances in Neural Information</i>	Jake McNeil, David Medina, Aalok Mehta, Jacob	937
879	<i>Processing Systems 35: Annual Conference on Neu-</i>	Menick, Luke Metz, Andrey Mishchenko, Pamela	938
880	<i>ral Information Processing Systems 2022, NeurIPS</i>	Mishkin, Vinnie Monaco, Evan Morikawa, Daniel	939
881	<i>2022, New Orleans, LA, USA, November 28 - Decem-</i>	Mossing, Tong Mu, Mira Murati, Oleg Murk, David	940
882	<i>ber 9, 2022</i> .	Mély, Ashvin Nair, Reiichiro Nakano, Rajeew Nayak,	941
883	Eric Mitchell, Charles Lin, Antoine Bosselut, Christo-	Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh,	942
884	pher D. Manning, and Chelsea Finn. 2022. Memory-	Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex	943
885	based model editing at scale . In <i>International Con-</i>	Paino, Joe Palermo, Ashley Pantuliano, Giambat-	944
886	<i>ference on Machine Learning, ICML 2022, 17-23</i>	tista Parascandolo, Joel Parish, Emy Parparita, Alex	945
887	<i>July 2022, Baltimore, Maryland, USA</i> , volume 162 of	Passos, Mikhail Pavlov, Andrew Peng, Adam Perel-	946
888	<i>Proceedings of Machine Learning Research</i> , pages	man, Filipe de Avila Belbute Peres, Michael Petrov,	947
889	15817–15831. PMLR.	Henrique Ponde de Oliveira Pinto, Michael, Poko-	948
890	OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal,	rny, Michelle Pokrass, Vitchyr H. Pong, Tolly Pow-	949
891	Lama Ahmad, Ilge Akkaya, Florencia Leoni Ale-	ell, Alethea Power, Boris Power, Elizabeth Proehl,	950
892	man, Diogo Almeida, Janko Altmenschmidt, Sam Alt-	Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh,	951
893	man, Shyamal Anadkat, Red Avila, Igor Babuschkin,	Cameron Raymond, Francis Real, Kendra Rimbach,	952
894	Suchir Balaji, Valerie Balcom, Paul Baltescu, Haim-	Carl Ross, Bob Rotsted, Henri Roussez, Nick Ry-	953
895	ing Bao, Mohammad Bavarian, Jeff Belgum, Ir-	der, Mario Saltarelli, Ted Sanders, Shibani Santurkar,	954
896	wan Bello, Jake Berdine, Gabriel Bernadett-Shapiro,	Girish Sastry, Heather Schmidt, David Schnurr, John	955
897	Christopher Berner, Lenny Bogdonoff, Oleg Boiko,	Schulman, Daniel Selsam, Kyla Sheppard, Toki	956
898	Madeline Boyd, Anna-Luisa Brakman, Greg Brock-	Sherbakov, Jessica Shieh, Sarah Shoker, Pranav	957
899	man, Tim Brooks, Miles Brundage, Kevin Button,	Shyam, Szymon Sidor, Eric Sigler, Maddie Simens,	958
900	Trevor Cai, Rosie Campbell, Andrew Cann, Brittany	Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin	959
901	Carey, Chelsea Carlson, Rory Carmichael, Brooke	Sokolowsky, Yang Song, Natalie Staudacher, Fe-	960
902	Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully	lipe Petroski Such, Natalie Summers, Ilya Sutskever,	961
903	Chen, Ruby Chen, Jason Chen, Mark Chen, Ben	Jie Tang, Nikolas Tezak, Madeleine B. Thompson,	962
904	Chess, Chester Cho, Casey Chu, Hyung Won Chung,	Phil Tillet, Amin Tootoonchian, Elizabeth Tseng,	963
905	Dave Cummings, Jeremiah Currier, Yunxing Dai,	Preston Tuggle, Nick Turley, Jerry Tworek, Juan Fe-	964
906	Cory Decareaux, Thomas Degry, Noah Deutsch,	lipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya,	965
907	Damien Deville, Arka Dhar, David Dohan, Steve	Chelsea Voss, Carroll Wainwright, Justin Jay Wang,	966
908	Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti,	Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei,	967
909	Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix,	CJ Weinmann, Akila Welihinda, Peter Welinder, Ji-	968
910	Simón Posada Fishman, Juston Forte, Isabella Ful-	ayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner,	969
911	ford, Leo Gao, Elie Georges, Christian Gibson, Vik	Clemens Winter, Samuel Wolrich, Hannah Wong,	970
		Lauren Workman, Sherwin Wu, Jeff Wu, Michael	971
		Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qim-	972
		ing Yuan, Wojciech Zaremba, Rowan Zellers, Chong	973
		Zhang, Marvin Zhang, Shengjia Zhao, Tianhao	974
		Zheng, Juntang Zhuang, William Zhuk, and Barret	975

976	Zoph. 2023. Gpt-4 technical report .	
977	Stephen Robertson, Hugo Zaragoza, et al. 2009. The	
978	probabilistic relevance framework: Bm25 and be-	
979	yond. <i>Foundations and Trends® in Information Re-</i>	
980	<i>trieval</i> , 3(4):333–389.	
981	Yucheng Shi, Qiaoyu Tan, Xuansheng Wu, Shaochen	
982	Zhong, Kaixiong Zhou, and Ninghao Liu. 2024.	
983	Retrieval-enhanced knowledge editing in language	
984	models for multi-hop question answering .	
985	Weihang Su, Yiran Hu, Anzhe Xie, Qingyao Ai, Zibing	
986	Que, Ning Zheng, Yun Liu, Weixing Shen, and Yiqun	
987	Liu. 2024a. Stard: A chinese statute retrieval dataset	
988	with real queries issued by non-professionals .	
989	Weihang Su, Yichen Tang, Qingyao Ai, Changyue	
990	Wang, Zhijing Wu, and Yiqun Liu. 2024b. Mitigating	
991	entity-level hallucination in large language models.	
992	<i>arXiv preprint arXiv:2407.09417</i> .	
993	Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu,	
994	Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024c. Un-	
995	supervised real-time hallucination detection based on	
996	the internal states of large language models . In <i>Find-</i>	
997	<i>ings of the Association for Computational Linguistics</i>	
998	<i>ACL 2024</i> , pages 14379–14391, Bangkok, Thailand	
999	and virtual meeting. Association for Computational	
1000	Linguistics.	
1001	Weihang Su, Changyue Wang, Anzhe Xie, Qingyao Ai,	
1002	Yiran Hu, and Yiqun Liu. 2024d. Legalaidd: A large	
1003	language model for the chinese legal field. https:	
1004	/github.com/oneal2000/LegalAID .	
1005	Qwen Team. 2024. Qwen2.5: A party of foundation	
1006	models .	
1007	Changyue Wang, Weihang Su, Yiran Hu, Qingyao Ai,	
1008	Yueyue Wu, Cheng Luo, Yiqun Liu, Min Zhang, and	
1009	Shaoping Ma. 2024a. Lekube: A knowledge up-	
1010	date benchmark for legal domain . In <i>Proceedings of</i>	
1011	<i>the 2024 Annual International ACM SIGIR Confer-</i>	
1012	<i>ence on Research and Development in Information</i>	
1013	<i>Retrieval in the Asia Pacific Region</i> , pages 175–185.	
1014	Changyue Wang, Weihang Su, Hu Yiran, Qingyao Ai,	
1015	Yueyue Wu, Cheng Luo, Yiqun Liu, Min Zhang, and	
1016	Shaoping Ma. 2024b. Lekube: A legal knowledge	
1017	update benchmark .	
1018	Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao,	
1019	Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan	
1020	Cheng, Kangwei Liu, Guozhou Zheng, et al. 2023a.	
1021	Easyedit: An easy-to-use knowledge editing frame-	
1022	work for large language models . <i>ArXiv preprint</i> ,	
1023	abs/2308.07269 .	
1024	Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V.	
1025	Le, Ed H. Chi, Sharan Narang, Aakanksha Chowd-	
1026	hery, and Denny Zhou. 2023b. Self-consistency	
1027	improves chain of thought reasoning in language	
1028	models . In <i>The Eleventh International Conference</i>	
1029	<i>on Learning Representations, ICLR 2023, Kigali,</i>	
1030	<i>Rwanda, May 1-5, 2023</i> . OpenReview.net.	
	Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xi-	1031
	aojian Ma, and Yitao Liang. 2024c. Rat: Re-	1032
	trieval augmented thoughts elicit context-aware	1033
	reasoning in long-horizon generation . <i>Preprint</i> ,	1034
	arXiv:2403.05313 .	1035
	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten	1036
	Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le,	1037
	and Denny Zhou. 2022. Chain-of-thought prompting	1038
	elicits reasoning in large language models . In <i>Ad-</i>	1039
	<i>vances in Neural Information Processing Systems 35:</i>	1040
	<i>Annual Conference on Neural Information Process-</i>	1041
	<i>ing Systems 2022, NeurIPS 2022, New Orleans, LA,</i>	1042
	<i>USA, November 28 - December 9, 2022</i> .	1043
	Thomas Wolf, Lysandre Debut, Victor Sanh, Julien	1044
	Chaumond, Clement Delangue, Anthony Moi, Pier-	1045
	ric Cistac, Tim Rault, Remi Louf, Morgan Funtow-	1046
	icz, Joe Davison, Sam Shleifer, Patrick von Platen,	1047
	Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu,	1048
	Teven Le Scao, Sylvain Gugger, Mariama Drame,	1049
	Quentin Lhoest, and Alexander Rush. 2020. Trans-	1050
	formers: State-of-the-art natural language processing .	1051
	In <i>Proceedings of the 2020 Conference on Empirical</i>	1052
	<i>Methods in Natural Language Processing: System</i>	1053
	<i>Demonstrations</i> , pages 38–45, Online. Association	1054
	for Computational Linguistics.	1055
	An Yang, Baosong Yang, Binyuan Hui, Bo Zheng,	1056
	Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan	1057
	Li, Dayiheng Liu, Fei Huang, Guanting Dong, Hao-	1058
	ran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian	1059
	Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin	1060
	Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang	1061
	Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang,	1062
	Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng	1063
	Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin,	1064
	Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu,	1065
	Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng,	1066
	Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin	1067
	Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang	1068
	Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu	1069
	Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2	1070
	technical report . <i>ArXiv preprint</i> , abs/2407.10671 .	1071
	Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio,	1072
	William Cohen, Ruslan Salakhutdinov, and Christo-	1073
	pher D. Manning. 2018. HotpotQA: A dataset for	1074
	diverse, explainable multi-hop question answering .	1075
	In <i>Proceedings of the 2018 Conference on Empiri-</i>	1076
	<i>cal Methods in Natural Language Processing</i> , pages	1077
	2369–2380, Brussels, Belgium. Association for Com-	1078
	putational Linguistics.	1079
	Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei	1080
	Qin, and Lidong Bing. 2023. Verify-and-edit: A	1081
	knowledge-enhanced chain-of-thought framework .	1082
	In <i>Proceedings of the 61st Annual Meeting of the</i>	1083
	<i>Association for Computational Linguistics (Volume</i>	1084
	<i>1: Long Papers)</i> , pages 5823–5840, Toronto, Canada.	1085
	Association for Computational Linguistics.	1086
	Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong	1087
	Wu, Jingjing Xu, and Baobao Chang. 2023. Can	1088
	we edit factual knowledge by in-context learning?	1089

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 Details of Benchmarks

Table 6 shows the statistics of the three benchmarks, including the total number of edits and queries.

D Details of Baselines

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

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

Table 6: Statistics on the number of edits and questions for the benchmarks

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

Algorithm 1 EditCoT Framework

```
1: 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  $F$   
2: Output: Final Answer  $A_{\text{final}}$   
3:  $A_{\text{old}} \leftarrow \text{GenerateAnswer}(M, q)$  ▷ Generate initial answer  
4:  $CoT_{\text{old}} \leftarrow \text{GenerateCoT}(M, q, A_{\text{old}})$  ▷ Generate initial chain-of-thought  
5: for  $i = 1$  to  $N$  do  
6:   for each step  $s$  in  $CoT_{\text{old}}$  do  
7:      $F_{\text{new}} \leftarrow \text{RetrieveFacts}(s, K)$  ▷ Retrieve new fact from memory  
8:     if  $F$  exists then  
9:        $F_{\text{new}} \leftarrow F[F_{\text{new}}]$  ▷ Map old fact to new fact if applicable  
10:    end if  
11:    if  $F_{\text{new}} \neq \emptyset$  then  
12:      break  
13:    end if  
14:  end for  
15:   $is\_Conflict \leftarrow \text{Verify}(M, q, CoT_{\text{old}}, F_{\text{new}})$  ▷ Verify conflict between CoT and new fact  
16:  if not  $is\_Conflict$  then  
17:    break  
18:  end if  
19:   $CoT_{\text{new}} \leftarrow \text{EditChain}(M_{\text{edit}}, q, CoT_{\text{old}}, F_{\text{new}})$  ▷ Edit chain-of-thought with new fact  
20:   $CoT_{\text{old}} \leftarrow CoT_{\text{new}}$  ▷ Update old CoT with new one  
21: end for  
22:  $A_{\text{final}} \leftarrow \text{GenerateAnswer}(M, CoT_{\text{new}})$  ▷ Generate final answer based on edited CoT  
23: return  $A_{\text{final}}$ 
```

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., 2024): 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 imple-

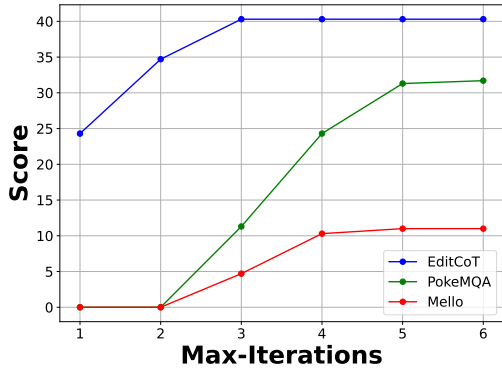


Figure 6: Performance with different max iterations. A full retrieval-generation cycle is defined as one iteration.

mented by Huggingface Transformers library (Wolf et al., 2020).

E 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.

F 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.

G 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.

H 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.

H.1 Prompt Templates of Dataset Construction

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

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.	
★ 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	★ 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>
	★ 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.

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:

1300
1301

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}
{...}

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:

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:

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:

H.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.

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

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:

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.

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

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 Moneyball a 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 'This 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 'Don'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]:

I 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.