EIKEA:Enhancing In-Context Knowledge Editing by Agents

Zibo Xu XUZB24@MAILS.JLU.EDU.CN

Xin Wang* XINWANG@JLU.EDU.CN
School of Artificial Intelligence, Jilin University

Editors: Hung-yi Lee and Tongliang Liu

Abstract

Recent knowledge editing methods have predominantly concentrated on modifying structured triplet knowledge within large language models. Compared to triplet-based knowledge, unstructured knowledge contains richer and more interrelated information, which increases the difficulty of editing. When relying solely on parameter-based editing methods, similar knowledge may interfere with each other due to their semantic overlap. Although previous studies have shown that directly applying in-context editing to unstructured knowledge with better results than parameter-based approaches, there is still considerable room for improvement. Previous studies have found that large language models are highly sensitive to the sequence of long text information,even the core content of the text may be masked due to positional influence. This indicates that, after rewriting unstructured facts, LLMs(Large Language Models) are better able to process and utilize the rewritten facts than the original facts. Inspired by this idea, we propose **EIKEA(Enhancing In-Context Knowledge Editing by Agents)**, a novel method that combines rewriting agent with IKE (In-Context Knowledge Editing), enabling language models to effectively internalize unstructured factual updates without modifying model parameters. We conduct comprehensive experiments on the WIKIUPDATE subset of the AKEW benchmark, demonstrating that our method significantly improves editing accuracy over baseline IKE and parameter-editing methods. Our method provides a practical, lightweight, and scalable solution to unstructured knowledge editing.

Keywords: Knowledge Editing; In-Context Learning; Unstructured Knowledge; Prompt Engineering; Language Models

1. Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing tasks. However, a persistent limitation lies in their static nature: once trained, they are unable to update or revise factual knowledge without undergoing expensive retraining or fine-tuning procedures. This limitation becomes particularly critical when the model must adapt to real-world changes, such as updates to political leadership, evolving scientific facts, or temporal events documented in natural language. In response, the field of knowledge editing has emerged, aiming to facilitate efficient and precise updates to the knowledge embedded within pretrained language models, thereby enhancing their adaptability to new or evolving information(Yao et al., 2023). Although several approaches have shown promising results in editing structured knowledge(Zhang et al., 2024), typically represented as subject-relation-object triplets, handling

* Corresponding author.

unstructured knowledge remains a significant challenge. Unstructured knowledge is often lengthy, noisy, and context-dependent, making it difficult for editing methods to isolate and update relevant information(Wu et al., 2024). From figure 1,We find that the current knowledge editing methods perform well only on structured facts, while struggling to handle unstructured knowledge and extracted triplets, which often contain significant amounts of redundant information.

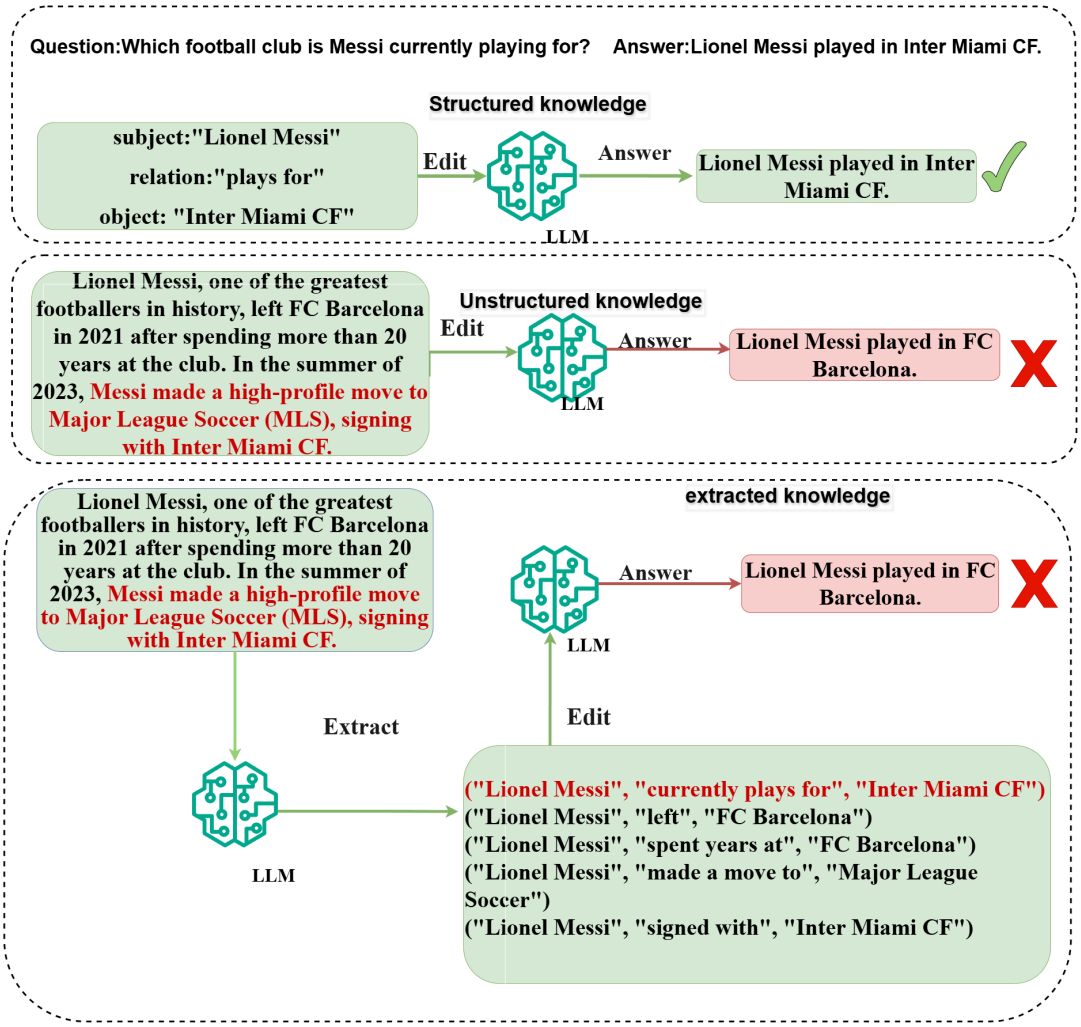

Figure 1: Showcase the overview of knowledge editing data formats.

Knowledge editing techniques are generally divided into two principal approaches: parameter-update strategies and in-context learning frameworks. Recent methods such as KN(Dai et al., 2022),ROME(Meng et al., 2022a), MEMIT (Meng et al., 2022b), and MEND (Mitchell et al., 2022b) attempt to localize and modify internal parameters in specific layers of the model. Other methods, including IKE(Zheng et al., 2023) , explore parameter-free strategies by performing in-context learning (ICL) using factual demonstrations. However, when

applied to unstructured textual updates, these methods suffer from significant performance degradation (Wu et al., 2024), suggesting that unstructured inputs inherently reduce the model's ability to absorb and generalize edited knowledge.Parameter-update methods generally require the factual input to be presented in a highly structured format, such as triplet-based knowledge representations. Extracting knowledge from paragraphs often results in semantically similar triplets that may obscure each other within the model's parameter space, thus reducing editing accuracy. While in-context learning methods like IKE(In-Context Knowledge Editing) harness the contextual competence of language models, they often fail to precisely target the core knowledge, which our method is designed to address. **EIKEA**(Enhancing In-Context Knowledge Editing by Agents) employs agent-based rewriting to foreground core knowledge while preserving peripheral content within paragraph-level information, thereby enhancing the accuracy of the knowledge editing process.

To address this gap, we propose a lightweight and scalable framework that enhances in-context knowledge editing by introducing a structured rewriting phase. Specifically, we design a CRI agent—a prompt-based rewriter that transforms noisy, unstructured knowledge into a semantically aligned format tailored for IKE. Inspired by recent advances in prompt engineering (Wei et al., 2022; Yao et al., 2022), our method leverages structured prompt templates to clarify, contextualize, and instruct the model on how to interpret and apply the new knowledge. Crucially, our approach avoids modifying model parameters, making it model-agnostic and deployment-friendly.

While our method is inspired by recent advances in prompt engineering and agent-style prompting, we do not employ autonomous multi-turn agents. Instead, we design static, structured prompt templates that encode role, reflection, and instruction signals to systematically rewrite unstructured knowledge for more effective in-context editing. This approach builds upon findings from recent studies in prompt engineering (Schick et al., 2023; Zhou et al., 2023), which demonstrate that prompt structure significantly affects model behavior.

We evaluate our method on the WIKIUPDATE benchmark , part of the AKEW dataset(Wu et al., 2024), and demonstrate substantial improvements over baseline IKE, fine-tuning, and editing-based approaches across several architectures including LLaMA2-7B and Qwen-7B. Our results indicate that rewriting unstructured knowledge significantly improves both editing success rates and downstream generalization. This highlights the utility of prompt-level intervention in the broader context of knowledge editing.

The main contributions of our work are:

(1) A novel framework that combines structured rewriting with in-context knowledge editing to effectively handle unstructured knowledge updates.A CRI agent design that systematically transforms unstructured inputs into formats better suited for knowledge editing
(2) Comprehensive experiments demonstrating significant improvements in editing accuracy and generalization across multiple model architectures
(3) Analysis of how different components of our approach contribute to overall performance through detailed ablation studies.Furthermore, the underlying causes of the observed phenomena were analyzed.

2. Related Work

Knowledge editing aims to update factual knowledge in language models without full retraining. Recent advancements in knowledge editing can be broadly categorized into three groups: methods that preserve original parameters, methods that locate and edit specific parameters, and methods that directly modify parameters.

2.1. Parameter-Preserving Methods

One group of methods focuses on augmenting the model with additional parameters, while the other involves embedding knowledge into the in-context learning (ICL) paradigm.These approaches focus on either introducing additional parameters or leveraging in-context learning capabilities:

Additional Parameter Methods: SERAC (Mitchell et al., 2022a) employs a classifier to detect editing needs and applies counterfactual models when necessary. **T-Patcher** (Huang et al., 2023) specifically trains neurons that are only activated on edited samples. **GRACE** (Hartvigsen et al., 2023) maintains fixed model weights while generating a local editing codebook.

In-Context Learning Methods: IKE reformulates editing as an in-context learning problem, utilizing memory and demonstration-based reasoning for factual updates. Our work builds upon this approach, enhancing it with structured rewriting strategies.

2.2. Locate-Then-Edit Approaches

A distinct approach employs a locate-and-edit methodology, identifying the pertinent parameters tied to the target knowledge and modifying them directly to implement the desired editing outcomes.These methods first identify internal knowledge representations before making targeted updates: **KN** focuses on editing through knowledge neurons. **ROME** performs rank-one editing at targeted transformer layers. **MEMIT** extends ROME's capabilities to handle batch editing scenarios. **UnKE** (Deng et al., 2025)is intended for unstructured knowledge, facilitating editing via non-local block key-value storage and cause-driven optimization. Such approaches assume knowledge resides in MLP layers, using term-oriented editing to modify behavior.

2.3. Direct Parameter Modification

Early approaches focused on direct fine-tuning with constraints to mitigate forgetting, such as FT (Zhu et al., 2020) and LoRA (Hu et al., 2021).Following this line of thought, a variety of other methods allow for knowledge editing by altering model parameters directly, without requiring explicit identification or positioning. More recent methods include:**MEND** utilizes gradient decomposition for rapid editing. **StableKE** (Wei et al., 2024) incorporates external knowledge during fine-tuning for stable editing

2.4. Challenges in Unstructured Settings

While most prior work evaluates on structured knowledge edits, AKEW(Wu et al., 2024) introduces a realistic benchmark that includes unstructured knowledge updates from real

Wikipedia pages. WIKIUPDATE dataset highlights the limitations of existing methods under natural language complexity. Traditional approaches face significant challenges in unstructured settings due to their reliance on strict localization or token-level edits.

However, when dealing with complex paragraph-level unstructured knowledge, these methods face serious challenges. Unstructured knowledge often involves densely connected content, where interference among overlapping information can hinder the accurate identification and editing of specific knowledge elements. As a result, such methods tend to perform poorly on datasets that involve extracted triplets from unstructured texts.Given these constraints, Our method builds on these insights while focusing on rewriting strategies to better align unstructured facts with the model's in-context reasoning capacity.

3. Methodology

Prior work shows AR models are sequence-sensitive in in-context learning (ICL), limiting performance and generalization. Our method addresses unstructured knowledge editing via two stages: CRI-based rewriting and in-context editing.Figure 2 illustrates the overall architecture of our approach, showing how the CRI components interact with the in-context knowledge editing process.

3.1. Task Formulation

Given an unstructured paragraph p that contains a target fact $f = (x^*, y^*)$, we convert it into k in-context demonstrations $\mathcal{C} = \{c_1, \ldots, c_k\}$ to inject into a language model M without updating its parameters.

$$y^* = \arg\max_y P_M(y \mid x, f, \mathcal{C}) \quad \text{for } x \in \mathcal{D}_{x^*},$$
$$\text{s.t.} \quad D_{\mathrm{KL}}\big(P_M(y \mid x, f, \mathcal{C}) \,\big\|\, P_M(y \mid x)\big) < \epsilon, \quad \forall x \notin \mathcal{D}_{x^*}.$$

for all $x \in \mathcal{D}_{x^*}$, where $\mathcal{D}_{x^*}$ is the set of prompts semantically related to x^*. This corresponds to the *accuracy goal*—successful application of the edited knowledge in relevant contexts. y^*denotes the ground-truth answer.

To evaluate the preservation of general capabilities, we assess M on a standard benchmark $\mathcal{T}_{\mathrm{MMLU}}$(Hendrycks et al., 2021), and compute the accuracy difference:

$$\Delta_{\mathrm{MMLU}} = \mathrm{Acc}(M_{\mathrm{edited}}, \mathcal{T}_{\mathrm{MMLU}}) - \mathrm{Acc}(M_{\mathrm{original}}, \mathcal{T}_{\mathrm{MMLU}}).$$

A smaller $|\Delta_{\mathrm{MMLU}}|$ indicates better preservation of the model's general capabilities.

3.2. CRI Agents Design

The CRI agent is designed to transform unstructured knowledge into a semantically aligned format designed for ICL. The agent consists of three components: Capacity, Reflection, and Instruction.

3.2.1. Capacity Component

The capacity component is designed to enhance the model's ability to understand complex knowledge, thus facilitating the development of the subsequent rewriting process.EIKEA

uses **You are a text rewriter focused on making information clear and understandable.** as the template prompt in this component.

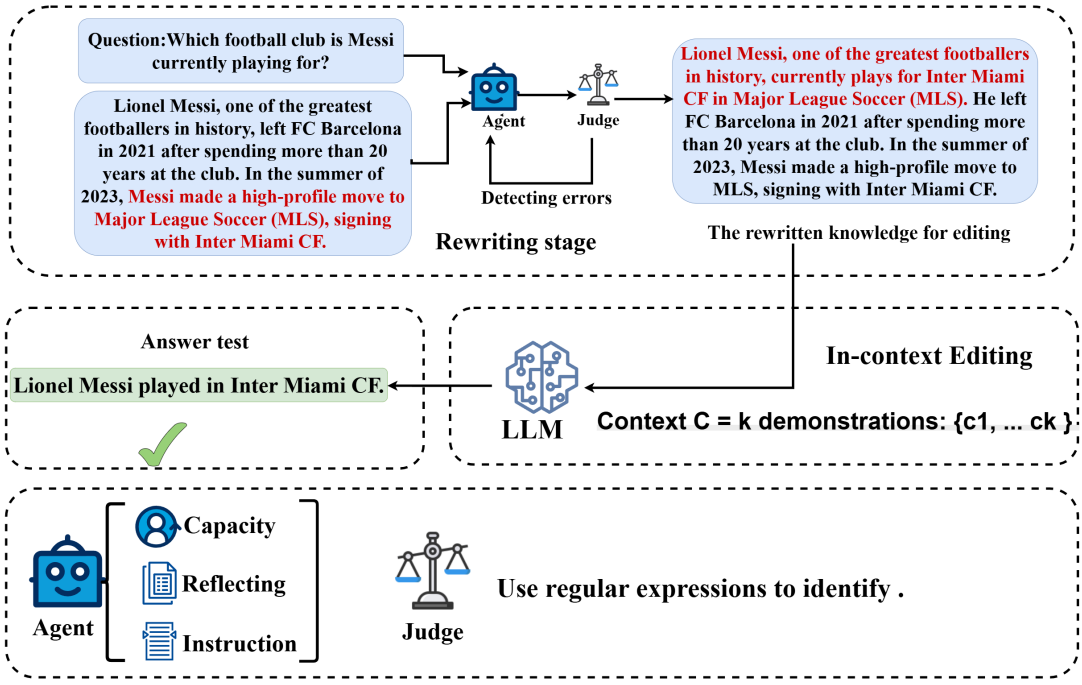

Figure 2: Architecture of the CRI framework for unstructured knowledge editing. The framework consists of two main phases: (1) CRI-based rewriting, which transforms unstructured input through Capacity, Reflection, and Instruction components (2) In-Context Knowledge Editing, which performs the actual knowledge update through demonstration-based learning. Within this process, the **Judge** module is designed to evaluate the plausibility and coherence of the rewritten text generated by CRI. If the output is assessed as inadequate, the rewriting procedure is iteratively invoked. In cases where the rewritten text fails to satisfy the evaluation criteria after two iterations, the system reverts to utilizing the original text for subsequent In-Context knowledge editing, thereby ensuring the robustness of the editing pipeline.

3.2.2. Reflection Component

The Reflection component is responsible for understanding the relationship between the input knowledge and the internal knowledge of the model. In the process of information generation by the model, the emergence of meaningless and redundant noise is inevitable. The Reflection Component serves to mitigate this by restricting the rewritten information to the original semantic scope of the unstructured knowledge, thereby ensuring that the generated content maintains precision to its intended meaning and context.EIKEA utilizes the statement **The text needs to be rewritten to help answer the question while**

preserving all relevant information as a template prompt to explicitly constrain text rewriting, ensuring that the related content related to the question is retained.

3.2.3. Instruction Component

The Instruction component is responsible for generating a structured prompt that clarifies, contextualizes, and instructs the model on how to interpret and apply the new knowledge. Unstructured knowledge, particularly when complex, encompasses a substantial amount of information. However, only a portion of this information is pertinent to answer a specific question. As such, existing methods for unstructured knowledge editing often treat the question as an attribute to be incorporated into the process. For example, the UnKE method implicitly represents the question as part of causal driving within the internal layers of the large model. In contrast, our approach uses **Please rewrite the following text to help answer this question: {question}** as the template prompt ,explicitly integrates the question as content within the Instruction Component, thereby ensuring that the rewritten knowledge emphasizes the critical, relevant elements of the information.

3.3. Knowledge Editing Workflow

To address the challenge of unstructured knowledge editing, we propose a two-stage agent-judge framework. An agent, termed CRI (Capacity–Reflection–Instruction), takes as input the unstructured paragraph p and question q, and rewrites them into a structured prompt $\tilde{p} = \mathrm{CRI}(p, q)$. After rewriting, the structured prompt $\tilde{p}$ is passed to a judgment module:

$$\mathcal{J}(\tilde{p}) \in \{\text{accept}, \text{reject}\},$$

which evaluates whether $\tilde{p}$ contains excessive noise or incorrect factual information. If rejected, the input is returned to the CRI agent for a second rewriting. After two failed attempts, we fall back to using the original input pair (q, p) directly.

During inference, the model M is frozen and conditioned on the accepted prompt:

$$\hat{y} = \arg\max_{y} P_M(y \mid x, \tilde{p}),$$

enabling factual adaptation through in-context knowledge editing without parameter updates.

Our method employs a robust rewriting approach that leverages the agent's ability to restructure complex unstructured facts. By placing core knowledge at the beginning of the rewritten text, the method improves clarity and facilitates downstream processing. The agent uses the same model type as the editor, ensuring better alignment with the model's reasoning process. The in-context editing process treats the CRI agent's output as a contextual prompt, enabling the integration of new knowledge without modifying the model's internal parameters. During inference, the model conditions its responses on this prompt, allowing it to adapt flexibly to updated factual content while preserving the integrity of the pre-trained architecture.

4. Experiment

4.1. Dataset

We evaluated our method in the WIKIUPDATE subset of AKEW . This dataset contains real-world knowledge updates from Wikipedia, structured in a comprehensive format that includes:Structured Fact,Unstructured Fact and Extracted Triplets.In our experiments, we simulate real-world knowledge editing scenarios by directly utilizing unstructured fact.

4.2. Experimental Setup

Base Language Models:
LLaMA2-7B, Qwen-7B, GPT-J, Mistral-7B,GPT-2XL.(Owing to the unsatisfactory MMLU test performance of GPT-2XL and GPT-J, and the constraints of their context window, these models were excluded from the MMLU test phase.)
Knowledge Editing Methods:
FT, LoRA, IKE (single), IKE (all),and UnKE. The implementations of FT, LoRA, IKE are carried out using the EasyEdit (Wang et al., 2024).
Metrics:
Accuracy:For the WIKIUPDATE subset, we leverage editing accuracy that measures how many edits are successful after editing(Cao et al., 2021).
General Ability: We follow the code in MMLU (Hendrycks et al., 2021) to evaluate each MMLU sample. The purpose of conducting this test is to evaluate the model's general ability segmented knowledge editing and its capacity to retain the remaining knowledge.

4.3. Knowledge Editing Settings

All knowledge editing methods are capable of modifying language models using structured facts and extracted triples. In the case of unstructured facts, in-context learning methods offer a natural and effective means of handling them.We use unstructured facts as training inputs for FT and LoRA.The training procedure follows the same protocol as that used in the AKEW framework.

It is important to note that Locate-Then-Edit methods, such as ROME and MEMIT, are inherently limited in their applicability to unstructured factual inputs. These approaches rely on explicitly defined subject–relation–object triples to compute intermediate representations, such as causal effects, which are essential to their editing mechanisms. As a result, they are not directly compatible with unstructured knowledge, which lacks the formal structure required for such computations.

We use one edit at each time for all methods by default or specially denoted as (single).For IKE , we additionally use all edits at one time, denoted as (all), to test their scalebility since they are state-of-the-art editing methods.

In the MMLU test,we average the scores of five MMLU samples for each unstructured sample after knowledge editing, and finally average these scores across all unstructured samples.To ensure the stability of the results, the temperature parameter value is set to 0.01 when the model generates responses.

4.4. Layer-wise Representation

To better understand how prompt rewriting affects the internal processing of factual information, we conduct a layer-wise analysis of token prediction probability across the transformer layers of LLaMA2-7B.

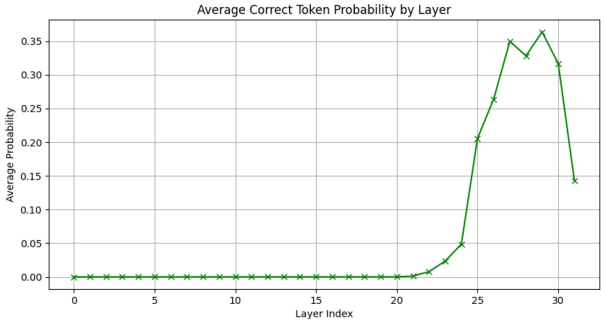

Figure 3: Average correct token probability across transformer layers. The probability sharply increases in the final layers, peaking around layers 28–31.

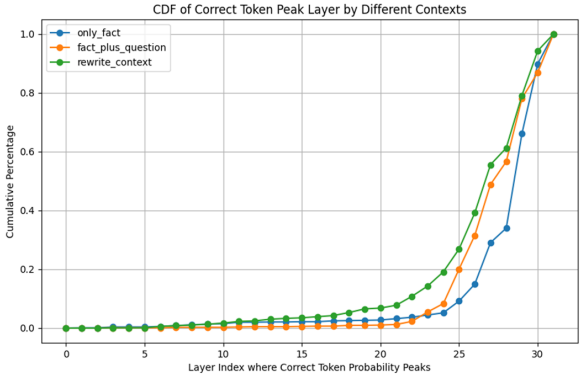

Figure 4: Cumulative distribution of the transformer layer at which correct token prediction probability peaks, under different context settings. Rewritten prompts (green) shift the convergence to earlier layers compared to raw facts (blue) or fact-plus-question pairs (orange).

Figure 3 shows that the correct token probability remains negligible in lower layers and begins rising steeply around layer 24, reaching its peak near layers 28–31.The decline in the final stage can be attributed to the common phenomenon of integrating knowledge as an output. As shown in Figure 4, rewritten prompts shift the factual convergence to earlier layers, thereby supporting the hypothesis that structured rewriting enhances internal token alignment. This outcome suggests that, compared to the original text and question piecing together fact, the model's rewritten version better facilitates comprehension, ultimately enhancing its interpretative capacity.

4.5. Results and Analysis

Table 1: Knowledge editing accuracy (%) on WIKIUPDATE's unstructured facts. The Noise column shows the percentage of edits that introduced unrelated changes.The symbol "–" indicates that the method is not compatible with the model.The results of FT and LoRA are directly adopted from the findings presented in the AKEW .

Method	LLaMA7B	Mistral7B	GPTj6B	GPT2xl	Qwen7B
Ours	**80.49**	**80.68**	**63.63**	**61.17**	**80.11**
UnKE	80.39	-	-	-	80.68
IKE(single)	60.03	57.95	26.99	22.44	68.56
FT	0.18	0.08	0.56	0.09	0.36
LoRA	0.94	0.04	5.16	5.44	1.23
IKE(all)	50.47	48.67	22.72	18.93	57.57

Table 2: MMLU accuracy (%) on different models. The values in parentheses indicate the performance drop relative to the base model. The symbol "-" indicates that the method is not applicable to the model.

Method	LLaMA7B	Qwen7B	Mistral7B
Base	48.11 (base)	55.55 (base)	57.54 (base)
Ours	**45.88** (-2.23)	55.40 (-0.15)	57.05 (-0.49)
UnKE	44.78 (-3.33)	54.57 (-0.98)	-
IKE	45.67 (-2.44)	**55.43** (-0.12)	**57.09** (-0.45)
LoRA	44.56 (-3.55)	52.48 (-3.07)	53.42 (-2.12)
FT	43.23 (-4.88)	51.32 (-4.23)	51.64 (-5.90)

Table 1 illustrates the comparative accuracy achieved by different knowledge editing methods when applied to unstructured factual inputs.Table 2 demonstrates that our method exerts a similarly minimal impact on the model's knowledge preservation as the IKE (in-context knowledge editing) method, which is notably lower compared to other approaches.According to these results,we have the following observations.

(1) **In-context knowledge editing is more effective than parameter-based editing methods for unstructured knowledge.** Parameter-based editing methods are typically implemented by selecting a specific layer within the model for modification. Prior research (Hase et al., 2023) has shown that the effectiveness of such edits is relatively insensitive to the exact storage location of the knowledge within the model. However, when applied to unstructured knowledge containing a large amount of information, these methods face notable challenges. Editing often fails, as dense interrelations among knowledge elements cause interference and semantic entanglement, leading to incorrect or unstable updates.

(2) **The processing of unstructured knowledge requires the disambiguation and extraction of its principal content.**Both our method and UNKE—a parameter-editing approach for unstructured knowledge—incorporate the question as an integral part of editing. Unstructured knowledge contains many complex, densely connected pieces that may

interfere during modification. Including the question anchors the process, guiding the model to focus on relevant segments and distill essential content for answering, thereby improving the effectiveness and precision of editing.
(3) **Model knowledge editing could impair its general reasoning performance.** Table 2 presents the MMLU test results before and after editing, clearly showing a decrease in the MMLU score post-editing, regardless of the method used. This reduction could be due to the editing process affecting the model's retained knowledge. Nonetheless, in-context knowledge editing, which relies on prompts, eliminates the possibility of damaging other knowledge.This reduction has been ruled out.The likely cause for the decline in the model's score is that the use of knowledge editing methods compromises the model's reasoning ability. This phenomenon has also been discussed in recent relevant research.(Nishi et al., 2025)

5. Ablation Study

We conduct comprehensive ablation studies to analyze the contribution of each component in our CRI(Capacity–Reflection–Instruction) framework. The experiments examine both individual components and their combinations, measuring their impact on IKE editing success rates in 1,056 test examples.

Table 3: Ablation study results showing IKE editing success rates under different component combinations. Numbers indicate successful edits out of 1056 test examples.

Component	Successful Edits	Success Rate (%)	Relative Gain (%)
Full (CRI)	852	80.68	100.0
CR	849	80.40	82.4
CI	844	79.92	52.9
RI	837	79.26	11.8
C	840	79.55	29.4
R	836	79.17	5.9
I	837	79.26	11.8
None	835	79.07	0.0

5.1. Key Findings

Our ablation analysis reveals several important insights:

> Based on the results of the ablation experiment, the components we designed have clear effects, with the most important being the **Capacity**, which means giving the model a clear role ensures excellent rewriting quality. The designs of the **Reflection** and **Instruction** sections are aimed at ensuring the retention of key information in complex paragraph content.

5.2. Impact Analysis

The relative gain of each component combination was calculated as:

$$\text{Relative Gain}(\%) = \frac{\text{Success}_{\text{variant}} - \text{Success}_{\text{base}}}{\text{Success}_{\text{full}} - \text{Success}_{\text{base}}} \times 100 \tag{1}$$

where $\text{Success}_{\text{base}} = 835$ (None) and $\text{Success}_{\text{full}} = 852$ (Full CRI).

These results demonstrate that:

(1) The full CRI framework improves editing success rate by 17 examples (1.61 percentage points) compared to the baseline

(2) CR combination captures 82.4% of the total possible improvement

(3) Each component contributes to the overall success, but their combination provides the most robust performance

(4) Even the baseline rewriting (without CRI) achieves a substantial success rate of 79.07%, indicating the fundamental effectiveness of our in-context editing approach

This analysis shows that while our base in-context editing method is effective, the CRI framework greatly improves unstructured knowledge updates. Combining all three components yields the highest success rate, highlighting the value of our comprehensive approach.

6. Conclusion and limitations

In this paper, we propose a method called **EIKEA** (Enhancing In-Context Knowledge Editing)to address the existing challenges of editing complex unstructured knowledge. EIKEA performs In-Context Knowledge Editing on the rewritten knowledge generated by the CRI agent. This design not only leverages the rewriting mechanism to retain information from unstructured knowledge, but also utilizes the large language model's contextual capabilities to edit unstructured knowledge effectively.

These findings open avenues for research in knowledge editing and show the potential of structured prompt engineering in enhancing language model capabilities. Our work lays a foundation for more sophisticated systems that handle the complexity and diversity of real-world information updates. While our CRI framework demonstrates strong performance in enhancing unstructured knowledge editing, several limitations and opportunities for future work remain:

The context window of large language models has a length limitation, which may result in degraded performance when processing very long unstructured updates that exceed the model's context window.

Our approach utilizes a rewriting mechanism to enhance text comprehension by the model, leveraging the context-learning capabilities of large language models and prompt engineering for knowledge editing. While this study investigates model probability analysis at the token level, it does not fully explore the mechanisms underlying context learning.After knowledge editing, large language models may experience a degradation in their reasoning capabilities. The underlying causes of this decline in reasoning performance and strategies for mitigating such effects remain key challenges that require further investigation and resolution.

Acknowledgments

This work was supported by a grant from the National Natural Science Foundation of China under grants (No.62372211, 62272191), and the Science and Technology Development Program of Jilin Province (No.20250102216JC).

References

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.emnlp-main.523.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022*, pages 8493–8502. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.581. URL https://doi.org/10.18653/v1/2022.acl-long.581.

Jingcheng Deng, Zihao Wei, Liang Pang, Hanxing Ding, Huawei Shen, and Xueqi Cheng. Everything is editable: Extend knowledge editing to unstructured data in large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*, 2025. URL https://openreview.net/forum?id=8UUvLQo7lY.

Trevor Hartvigsen, Yichi Li, and Greg Durrett. GRACE: generative rewriting and contextual editing for lifelong model editing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, pages 14254–14269. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.796. URL https://doi.org/10.18653/v1/2023.acl-long.796.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, pages 17643–17668, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/3927bbdcf0e8d1fa8aa23c26f358a281-Abstract-Conference.html.

Dan Hendrycks, Colton Burns, Samuel Basart, Siddhant Kadavath, Jacob Steinhardt, Sarthak Trivedi, Eric Wallace, Haotian Xu, and Daphne Ippolito. Mmlu: Massive multitask language understanding evaluation. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021. URL https://arxiv.org/abs/2109.08888.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

Zhen Huang, Yelong Shen, Xiaodong Zhang, Jie Zhou, Weiqing Rong, and Zhi Xiong. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023. URL https://openreview.net/forum?id=AQHtt3hHDg.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. ROME: 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*, pages 17359–17372, 2022a. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a35d91bbf13508e095552-Abstract-Conference.html.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. MEMIT: memory editing for memory-based inference in transformers. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, pages 32610–32623, 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1ea18f9b637856fe5c8b5a0cc7b0ae-Abstract-Conference.html.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. SERAC: memory-based model editing at scale. In *The Tenth International Conference on Learning Representations, ICLR 2022*, 2022a. URL https://openreview.net/forum?id=seqF4qyufb.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Fast model editing at scale. In *The Tenth International Conference on Learning Representations, ICLR 2022*, 2022b. URL https://openreview.net/forum?id=0DcZxeWfOPt.

Kento Nishi, Rahul Ramesh, Maya Okawa, Mikail Khona, Hidenori Tanaka, and Ekdeep Singh Lubana. Representation shattering in transformers: A synthetic study with knowledge editing. In *Proceedings of the 2025 International Conference on Machine Learning (ICML 2025)*, 2025. URL https://arxiv.org/abs/2410.17194.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *CoRR*, abs/2302.04761, 2023. doi: 10.48550/arXiv.2302.04761. URL https://doi.org/10.48550/arXiv.2302.04761.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. Easyedit: An easy-to-use knowledge editing framework for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 82–93. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-demos.9. URL https://aclanthology.org/2024.acl-demos.9.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 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*, pages 24824–24837, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Zihao Wei, Liang Pang, Hanxing Ding, Jingcheng Deng, Huawei Shen, and Xueqi Cheng. Stableke: Enhancing stability in knowledge editing for large language models. *CoRR*, abs/2402.13048, 2024. URL https://arxiv.org/abs/2402.13048.

Xiaobao Wu, Liangming Pan, William Yang Wang, and Anh Tuan Luu. AKEW: assessing knowledge editing in the wild. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024*, pages 15118–15133. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.843. URL https://doi.org/10.18653/v1/2024.emnlp-main.843.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *Nature Machine Intelligence*, 4(12):1371–1381, 2022. doi: 10.1038/s42256-022-00583-4. URL https://doi.org/10.1038/s42256-022-00583-4.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 10222–10240, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/V1/2023.EMNLP-MAIN.632. URL https://doi.org/10.18653/V1/2023.EMNLP-MAIN.632.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. A comprehensive study of knowledge editing for large language models. *CoRR*, abs/2401.01286, 2024. doi: 10.48550/ARXIV.2401.01286. URL https://doi.org/10.48550/arXiv.2401.01286.

Chunting Zheng, Lemao Li, Qian Dong, Yuxuan Fan, Zhen Wu, Jun Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning? *CoRR*, abs/2305.13233, 2023. doi: 10.48550/arXiv.2305.13233. URL https://doi.org/10.48550/arXiv.2305.13233.

Zhibin Zhou, Nuo Liu, Zhuohan Li, Xiangru Zheng, Tao Zhang, Saining Xie, Yuxin Wang, and Eric P. Xing. Rewoo: Decoupling reasoning from observations for efficient augmented language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 14228–14241. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.862. URL https://arxiv.org/abs/2305.18323.

Chen Zhu, Tom Grievink, James Kendall, Tom B. Brown, Benjamin Mann, Jared Kaplan, and Dario Amodei. Modifying memories in transformer models. *CoRR*, abs/2012.00363, 2020. URL https://arxiv.org/abs/2012.00363.

Appendix A: Rewriting Prompt Template

Capacity (Role)
You are a text rewriter focused on making information clear and understandable.

Reflection (Insight)
The text needs to be rewritten to help answer the question while preserving all relevant information.

Instruction (Statement)
Please rewrite the following text to help answer this question: "{question}"

Text: {context}

Style (Personality)
Be clear and focused on information relevant to the question.

Performance (Experiment)
REQUIREMENTS:
1. You MUST keep ALL information that is directly related to answering the question
2. You MUST make the text easier to understand
3. You MUST NOT remove any details that could help answer the question
4. You MUST NOT include any template or framework text
5. You MUST NOT add any information not in the original text

Please provide a clear and focused explanation that directly helps answer the question.

Appendix B: Detailed Experimental Results

Table 4: Detailed knowledge editing results including sample sizes for each method and model combination.

Method	LLaMA7B	Mistral7B	GPTj6B	GPT2xl	Qwen
Ours	80.49 (850/1056)	80.68 (852/1056)	63.63 (672/1056)	61.17 (646/1056)	80.11 (846/1056)
UnKE	80.39 (849/1056)	- -	- -	- -	80.68 (852/1056)
IKE(single)	60.03 (634/1056)	57.95 (612/1056)	26.99 (285/1056)	22.44 (237/1056)	68.56 (724/1056)
FT	0.18 (2/1056)	0.08 (1/1056)	0.56 (6/1056)	0.09 (1/1056)	0.36 (4/1056)
LoRA	0.94 (10/1056)	0.04 (0/1056)	5.16 (54/1056)	5.44 (57/1056)	1.23 (13/1056)
IKE(all)	50.47 (533/1056)	48.67 (514/1056)	22.72 (240/1056)	18.93 (200/1056)	57.57 (608/1056)

Table 4 presents complete experimental results, including successful edits over total attempts for each method-model pair on 1056 WIKIUPDATE test examples. A portion of the dataset—17.04%—contains answers that cannot be directly mapped, making such unstructured knowledge inherently uneditable and indicating that our results are constrained by this factor rather than the method itself.