Editing as Unlearning: Are Knowledge Editing Methods Strong Baselines for Large Language Model Unlearning?

Zexi Li[*†] University of Cambridge, Zhejiang University
Xiangzhu Wang[*] Zhejiang University
William F. Shen University of Cambridge
Meghdad Kurmanji University of Cambridge
Xinchi Qiu University of Cambridge
Dongqi Cai University of Cambridge
Chao Wu[†] Zhejiang University
Nicholas D. Lane[†] University of Cambridge

Abstract

Large language Model (LLM) unlearning, i.e., selectively removing information from LLMs, is vital for responsible model deployment. Differently, LLM knowledge editing aims to modify LLM knowledge instead of removing it. Though editing and unlearning seem to be two distinct tasks, we find there is a tight connection between them. In this paper, we conceptualize unlearning as a special case of editing where information is modified to a refusal or "empty set" $\emptyset$ response, signifying its removal. This paper thus investigates if knowledge editing techniques are strong baselines for LLM unlearning. We evaluate state-of-the-art (SOTA) editing methods (e.g., ROME, MEMIT, GRACE, WISE, and AlphaEdit) against existing unlearning approaches on pretrained and finetuned knowledge. Results show certain editing methods, notably WISE and AlphaEdit, are effective unlearning baselines, especially for pretrained knowledge, and excel in generating human-aligned refusal answers. To better adapt editing methods for unlearning applications, we propose practical recipes including self-improvement and query merging. The former leverages the LLM's own in-context learning ability to craft a more human-aligned unlearning target, and the latter enables ROME and MEMIT to perform well in unlearning longer sample sequences. We advocate for the unlearning community to adopt SOTA editing methods as baselines and explore unlearning from an editing perspective for more holistic LLM memory control.

1 Introduction

In recent years, large language models (LLMs) [37, 19, 2] have achieved remarkable success, with their broad knowledge enabling a wide range of applications, including mobile assistants [42], medical diagnosis [35], coding copilot [47]. However, as these models evolve, managing the knowledge they retain and generate has become increasingly critical. In particular, growing concerns around privacy [5], ethics [29], and legal compliance (such as with the General Data Protection Regulation (GDPR) [40] and the California Consumer Privacy Act (CCPA) [30]) have brought attention to the *"right to be forgotten"*, which grants individuals the legal right to request the deletion or modification of personal data. These factors highlight the growing need for mechanisms that enable LLMs

[*]Equal contributions.

[†]Correspondence: Zexi Li (zexi.li@zju.edu.cn), Chao Wu (chao.wu@zju.edu.cn), and Nicholas D. Lane (ndl32@cam.ac.uk).

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Lock-LLM: Prevent Unauthorized Knowledge Use from LLMs.

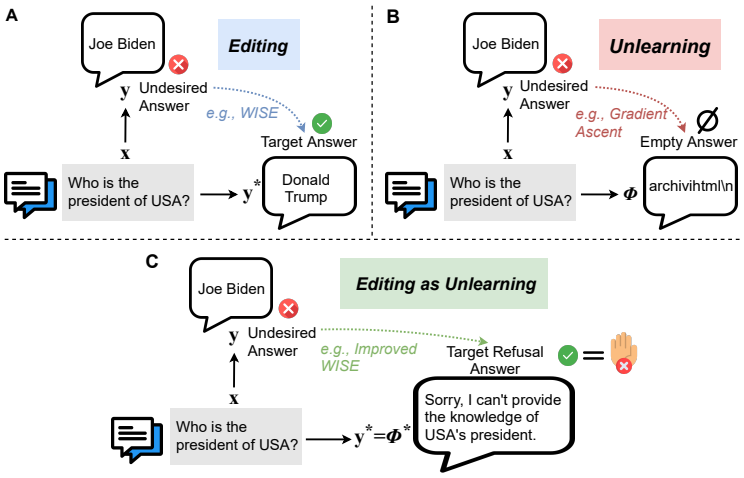

Figure 1: **Illustrations of the connection between editing and unlearning for LLMs. A:** Editing aims to alter the knowledge to a target. **B:** Unlearning tries to remove the knowledge and generate an "empty" (without information) answer. **C:** Editing as unlearning, can be done by editing that alters the knowledge into a target refusal answer.

to unlearn specific data points (i.e., instance-level knowledge), particularly sensitive or erroneous information, that may have been unintentionally incorporated during training. Failure to address this can lead to privacy violations, legal risks, and erosion of public trust, making effective unlearning a critical capability for responsible LLM deployment.

Instance-level knowledge unlearning (hereafter referred to as *unlearning*) is a complex task. It requires selectively removing specific knowledge from a model without affecting its overall performance. This is particularly challenging in the context of LLMs, which store vast amounts of data across billions of parameters. While traditional machine learning methods often focus on task-specific model updates [7, 28], LLM unlearning demands a more nuanced approach to prevent "catastrophic forgetting" and maintain the model's generalization capabilities.

Interestingly, the field of knowledge editing [51] (also known as *model editing*) — which involves modifying a model's knowledge, typically to correct or update information — shares inherent commonalities with unlearning. While unlearning focuses on removing the knowledge, knowledge editing aims to alter the knowledge, and both tasks require precise control over the model's stored knowledge. As shown in Figure 1, we find that removing knowledge is a special case of altering knowledge by replacing the targeted answer from y^* to $\emptyset$ (empty set). Since a successfully unlearned model should emulate the base model's behavior when presented with unseen data, the appropriate behavioral target is a contextualized expression of ignorance (hereafter referred to as a refusal answer), which mainstream instruction-tuned models are typically aligned to produce. Prior work refers to this behavioral fidelity as the *controllability of unlearning* [33]. As such, the refusal answer can be viewed as the $\emptyset$ knowledge of LLMs, which means that knowledge editing can inherently do unlearning as long as changing the target answer into a refusal. It may suggest that techniques from knowledge editing could provide a solid foundation for effective unlearning. Though some works have raised preliminary discussions about the connection between editing and unlearning [22, 53, 39], in the LLM unlearning community, we find that most of the technical papers may pay less attention than expected to knowledge editing, not implementing editing methods as baselines [50, 21, 17]. Meanwhile, the field of LLM knowledge editing is developing rapidly, facilitating classic and state-of-the-art (SOTA) methods like ROME [25], MEMIT [26], WISE [44], and AlphaEdit [6]. In addition, compared with vanilla finetuning, editing methods also have the merits of lightweight and efficiency [51]. However, LLM unlearning is at a more early stage, some existing baselines are borrowed from machine unlearning of vision classification tasks (e.g., GA and GD), not tailored to generative models like LLMs. This forces us to pose the following research question:

Can knowledge editing methods be strong baselines for LLM unlearning?

Therefore, this paper aims to provide a timely answer to the above question by investigating and evaluating classic and SOTA LLM editing methods for LLM unlearning. We hope this can bridge the gap between the two communities and provide some insights for future research. Specifically, we first study whether editing methods can unlearn as effectively as unlearning baselines for pretrained and finetuned knowledge. Then, we investigate the boundaries of editing methods for unlearning,

identifying the key challenges. Lastly, we propose some practical modules that can better adapt editing in unlearning tasks for future implications.

Our contributions are as follows.

- We bridge the gap between LLM editing and unlearning communities by investigating whether editing methods can serve as strong baselines for LLM unlearning.
- We explore two practical methods that can better adapt editing methods in unlearning tasks. The proposed self-improvement pipeline leverages the LLM's own in-context learning ability to craft a more human-aligned unlearning target, and the proposed query merging technique enables ROME and MEMIT to perform well in unlearning longer sample sequences.
- We advocate the LLM unlearning community to take the SOTA editing methods as unlearning baselines when conducting evaluation as well as to study unlearning from the knowledge editing perspective to gain a more holistic understanding of LLM memory control and knowledge mechanism.

Our takeaway findings are summarized as follows.

- We find some LLM editing methods, especially WISE and AlphaEdit *are strong baselines* especially when unlearning *pretrained knowledge*.
- We emphasize the importance of human value alignment of LLM unlearning, suggesting that LLMs should generate trustworthy refusal answers instead of random tokens or misleading phrases. We find some editing methods (i.e., WISE) have a dominant advantage on human value alignment over unlearning methods.
- Our proposed self-improvement pipeline for editing methods (e.g., WISE and AlphaEdit) that can potentially improve human value alignment as well as the generalization ability under rephrase-prompted attacks. Additionally, the proposed query merging technique can enable ROME and MEMIT to do unlearning well under long sequences, surpassing all the unlearning baselines.

2 Preliminaries

2.1 LLM Knowledge Editing

We give a definition of the LLM editing setup. Let $f_\Theta : \mathbb{X} \mapsto \mathbb{Y}$, parameterized by Θ, denote a model function mapping an input $\mathbf{x}$ to the prediction $f_\Theta(\mathbf{x})$. The initial model before editing is Θ_0, which is trained on a large corpus $\mathcal{D}_{\text{train}}$. When the LLM needs editing to alter some knowledge, it has an editing dataset as $\mathcal{D}^*_{\text{edit}} = \{(\mathcal{X}^*_e, \mathcal{Y}^*_e)|(\mathbf{x}_1, \mathbf{y}^*_1), ..., (\mathbf{x}_T, \mathbf{y}^*_T)\}$ which has a sequence or batch length of T. Given a query $\mathbf{x}_T$, the editing method maps the knowledge to the target as $\mathbf{y}_T \to \mathbf{y}^*_T$ where $\mathbf{y}_T$ is the previous knowledge. At editing, the updated LLM f_{Θ^*} is expected to satisfy:

$$f_{\Theta^*}(\mathbf{x}) = \begin{cases} \mathbf{y}^* & \text{if } \mathbf{x} \in \mathcal{X}^*_e, \\ f_{\Theta_0}(\mathbf{x}) & \text{if } \mathbf{x} \notin \mathcal{X}^*_e. \end{cases} \tag{1}$$

Equation 1 describes that after knowledge editing, the LLM should make the correct prediction of the edits while preserving the irrelevant and generic knowledge, especially general training corpus $\mathcal{D}_{\text{train}}$.

2.2 LLM Unlearning

Following the editing setup, we now consider the problem of LLM unlearning. It has a unlearning dataset $\mathcal{D}'_{\text{unlearn}} = \{(\mathcal{X}'_u, \mathcal{Y}'_u)|(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_T, \mathbf{y}_T)\}$ which is usually a part of the training data $\mathcal{D}_{\text{train}}$. Given the query $\mathbf{x}_T$, $\mathbf{y}_T$ is the ground-truth answer that is used in the training but needs to be forgotten. Ideally, after unlearning, the updated LLM model $f_{\Theta'}$ should satisfy:

$$f_{\Theta'}(\mathbf{x}) \begin{cases} \neq \mathbf{y} & \text{if } \mathbf{x} \in \mathcal{X}'_u, \\ = f_{\Theta_0}(\mathbf{x}) & \text{if } \mathbf{x} \notin \mathcal{X}'_u. \end{cases} \tag{2}$$

Equation 2 defines the unlearning objective: removing knowledge of the forget set $\mathcal{D}'_{\text{unlearn}}$ while preserving knowledge from the remaining data. To prevent catastrophic forgetting, some methods use a retain set or reference model. However, retain sets may be impractical in certain scenarios [46], and models should ideally preserve open-set knowledge. Ideally, the goal is for unlearning on $\mathcal{D}'_{\text{unlearn}}$ to approximate retraining from scratch on $\mathcal{D}_{\text{train}} \setminus \mathcal{D}'_{\text{unlearn}}$.

3 Methodology

3.1 Making Editing Applicable in Unlearning

Equations 1 and 2 have shown the inherent connections between editing and unlearning, and the key difference is the within-scope condition. Unlike classification models in vision tasks, LLMs as generative models, have the ability to refuse to answer as a form of removing the knowledge. Therefore,

assuming there is an "empty" set $\varnothing = \{\emptyset_1, ..., \emptyset_T\}$ which is the sentences telling the users that "I don't know", change the unlearning set $\mathcal{D}'_{\text{unlearn}}$ into $\mathcal{D}^*_{\text{edit-as-unlearn}} = \{(\mathcal{X}^*_{e2u}, \mathcal{Y}^*_{e2u})|(\mathbf{x}_1, \emptyset_1), ..., (\mathbf{x}_T, \emptyset_T)\}$. Applying the new dataset to editing methods, the objective of Equation 1 changes to:

$$f_{\Theta^*}(\mathbf{x}) = \begin{cases} \emptyset & \text{if } \mathbf{x} \in \mathcal{X}^*_{e2u}, \\ f_{\Theta_0}(\mathbf{x}) & \text{if } \mathbf{x} \notin \mathcal{X}^*_{e2u}. \end{cases} \tag{3}$$

Equation 3 bridges from editing to unlearning, making it applicable to verify whether editing methods are strong baselines for unlearning.

3.2 Improving Editing in Unlearning

Knowledge editing was not tailored for unlearning, as a result, it may have some limitations when directly being applied, e.g., different learning objectives and different sample lengths. Therefore, as shown in Figure 2, we explore some techniques to better adapt editing methods in unlearning.

Self-improvement pipeline. A good refusal answer from LLMs should be trustworthy and aligned with human values. We find if the editing target answers are random sentences from the vanilla "I don't know" set, it will let the LLMs generate answers that are less trustworthy, e.g., low generalization, misleading, or without entailing the entities mentioned in questions. Therefore, we craft a self-improvement pipeline to let LLMs create tailored refusal answers to each forget question before unlearning. Specifically, we provide instructions and exemplars to help LLMs generate more tailored unlearning targets for each question (for detailed prompts, see subsection C.2). Thanks to their in-context learning ability, LLMs can produce trustworthy answers that reflect the question's entities without misleading information. This helps them learn patterns between questions and refusal answers during the latter unlearning phase. The experiments in subsection 4.2 will show that the self-improvement pipeline can improve the answers regarding human value alignment and improve generalization under rephrased attacks.

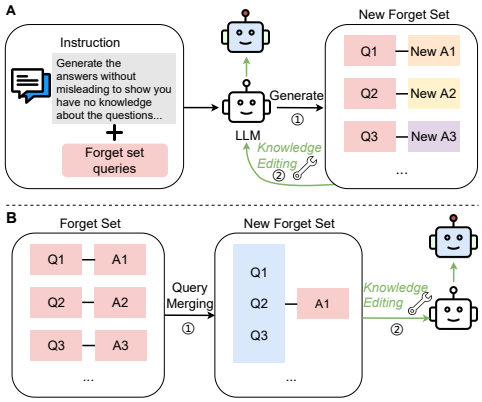

Figure 2: **Methods of improving editing algorithms in unlearning settings. A:** Self-improvement pipeline improves generalization and human value alignment for AlphaEdit and WISE. **B:** Query merging technique enables ROME and MEMIT to perform well under long unlearning sequences.

Query merging technique. Some locate-and-edit editing methods like ROME and MEMIT cannot well perform under long sequences of editing [10, 44], and this drawback still exists when editing applies to unlearning, which limits their broader application in unlearning. However, we find that, unlike the vanilla editing setting where every edit has one unique target answer, under the editing-as-unlearning setting, several forget queries can be mapped to a common refusal answer — the model can say the same "I don't know" to many queries. This inspires us the query merging technique that concatenates several queries into one and uses one refusal answer as the editing target. This simple technique can enable ROME and MEMIT to perform very well under unlearning, achieving obvious performance advantages over the unlearning baselines (Figure 3).

4 Empirical Results

In this section, we conduct experiments to address the following research questions:

- **RQ1:** Can editing methods outperform the unlearning baselines when unlearning the pretrained knowledge and the finetuned knowledge respectively? Which editing methods are most effective for unlearning tasks?
- **RQ2:** What are the comprehensive performances of the editing methods in unlearning? Can they perform well under rephrase attacks or with different numbers of forget samples?
- **RQ3:** How to improve editing methods for unlearning tasks? Can the editing methods generate better answers that align with human values than the unlearning baselines? Can we make some inapplicable editing methods (i.e., ROME and MEMIT) applicable and perform well for unlearning?

For the settings, due to page limit, please refer to the appendix.

Table 1: **Main results comparing editing and unlearning methods.** The number of forget samples in the factual dataset is 40 and PISTOL's is 20. The forget set performance corresponds to the *reliability* metric of editing and the retain set corresponds to *locality*. In some cases, particular methods will make LLMs non-functional (e.g., near-zero Rouge1 for both forget and retain sets) or without any forgetting, and we make these cases in gray. For every metric of each setting, we mark the best of unlearning and editing, respectively **in bold**, and we mark the Top 2 out of all methods in underline.

Dataset	**Factual dataset** (pretrained knowledge)															
Model	Llama2-7B								Mistral-7B							
Testset	Forget set (reliability)				Retain set (locality)				Forget set (reliability)				Retain set (locality)			
Metric	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑	Prob.↑	MRR↑	Hit-Rate↑	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑	Prob.↑	MRR↑	Hit-Rate↑
GA	0.00	0.59	0.00	0.00	0.00	0.52	0.00	0.00	0.00	0.62	0.06	0.09	0.00	0.56	0.02	0.06
GD	**0.30**	**0.36**	0.02	**0.02**	**0.62**	**0.27**	**0.12**	**0.13**	**0.00**	**0.56**	0.05	0.09	**0.52**	0.49	**0.18**	**0.54**
KL	0.00	0.55	0.00	0.00	0.00	0.48	0.00	0.00	0.00	0.42	0.06	0.08	0.00	0.43	0.02	0.06
DPO	0.36	**0.36**	**0.01**	**0.02**	0.45	**0.27**	0.03	0.04	0.03	**0.60**	**0.00**	**0.03**	0.43	**0.57**	0.07	0.15
ROME	0.01	0.41	0.01	0.01	0.04	0.32	0.01	0.01	0.00	0.54	0.04	0.06	0.00	0.48	0.02	0.04
MEMIT	0.02	0.82	0.00	0.00	0.01	0.78	0.00	0.00	–	–	–	–	–	–	–	–
GRACE	0.65	**0.35**	0.18	0.22	**0.82**	**0.26**	**0.21**	**0.26**	0.93	0.44	0.37	0.68	**0.82**	**0.45**	**0.34**	**0.69**
WISE	0.28	0.37	0.11	0.14	0.76	**0.26**	0.18	0.23	**0.05**	**0.13**	**0.01**	**0.08**	0.13	0.12	0.10	0.36
AlphaEdit	**0.08**	**0.35**	**0.04**	**0.05**	0.69	**0.26**	0.12	0.15	0.26	0.45	0.09	0.22	0.66	**0.45**	0.24	0.53
Dataset	**PISTOL** (finetuned knowledge)															
Model	Llama2-7B								Mistral-7B							
Testset	Forget set (reliability)				Retain set (locality)				Forget set (reliability)				Retain set (locality)			
Metric	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑	Prob.↑	MRR↑	Hit-Rate↑	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑	Prob.↑	MRR↑	Hit-Rate↑
GA	**0.16**	0.29	0.18	0.19	0.69	0.29	0.20	0.20	0.27	**0.54**	0.15	0.39	**0.76**	0.54	0.24	**0.59**
GD	0.25	0.29	0.17	0.17	0.80	0.29	0.20	0.20	0.22	0.58	0.16	**0.31**	**0.76**	**0.58**	**0.25**	0.56
KL	0.82	0.33	0.23	0.33	**0.98**	**0.33**	**0.26**	**0.36**	**0.08**	0.55	**0.05**	0.35	0.34	0.55	0.11	0.51
DPO	0.18	**0.28**	**0.15**	**0.15**	0.86	0.28	0.22	0.22	0.00	0.44	0.01	0.04	0.06	0.44	0.02	0.05
ROME	0.00	0.37	0.00	0.00	0.00	0.37	0.00	0.01	0.04	0.20	0.09	0.39	0.02	0.20	0.10	0.40
MEMIT	0.00	0.42	0.16	0.18	0.00	0.42	0.17	0.23	-	-	-	-	-	-	-	-
GRACE	1.00	0.28	0.25	0.25	1.00	0.29	0.22	0.22	1.00	0.48	0.33	0.81	1.00	0.48	0.31	0.78
WISE	0.68	**0.25**	0.26	0.27	**0.94**	0.25	**0.21**	**0.21**	**0.05**	**0.29**	**0.04**	**0.30**	**0.36**	0.29	0.12	0.41
AlphaEdit	**0.05**	0.28	**0.14**	**0.16**	0.25	**0.28**	0.15	0.17	**0.05**	0.47	0.14	0.47	0.12	**0.47**	**0.18**	**0.55**

4.1 General Performance of Editing Methods in Unlearning (RQ1)

We compare 4 unlearning methods and 5 editing methods under 4 settings and the results are in Table 1. The factual dataset from TOFU consists of the knowledge during LLM pretraining, and we test Rouge1 before unlearning: 0.82 for Llama2-7B and 0.86 for Mistral-7B. The PISTOL dataset focuses on structural unlearning under finetune-then-unlearn setup, and we finetune the base models on the whole PISTOL dataset to reach 1.0 Rouge1 and then forget a proportion of the finetuned set.

Ob1: Unlearning might lead to model failure, but some editing methods are more robust. Results in Table 1 show that some methods will result in the retain model non-usable post unlearning. This happens to unlearning methods GA and KL, as well as editing methods ROME and MEMIT. However, we will show later in Subsection 4.2 that with the query merging technique, ROME and MEMIT can produce excellent unlearning performances. Notably, WISE and AlphaEdit consistently perform well across all settings.

Ob2: Editing methods are strong baselines for unlearning, especially for pretrained knowledge. "Forget" and "Retain" is an important tradeoff in unlearning, some methods may unlearn too much, causing damage to general or retain knowledge. Therefore, we count the methods that get the Top-2 ranking for both forget and retain sets within the same setting, and they are GD, DPO, GRACE, and WISE for factual dataset and GA, GD, KL, DPO, and WISE for PISTOL. It seems that editing performs better on pretrained knowledge and basic unlearning methods perform better on finetuned knowledge. This might be owing to the inherently different knowledge mechanisms between pretraining and finetuning [4], and editing is naturally designed for altering the pretrained knowledge of LLMs. We note that unlearning pretrained knowledge is important for real practice since most of the factual knowledge is obtained during pretraining.

Table 2: **Results under rephrase attack (generalization).** Factual dataset, 40 forget samples, Llama2-7B.

Testset	Rephrased forget set (generalization)			
Metric	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓
GA	0.00	0.59	0.00	0.00
GD	**0.42** (0.12↑)	**0.34**	0.03 (0.01↑)	0.03 (0.01↑)
KL	0.00	0.54	0.00	0.00
DPO	0.52 (0.15↑)	**0.34**	**0.00**	**0.01**
ROME	0.01	0.40	0.01	0.01
MEMIT	0.00	0.83	0.00	0.00
GRACE	0.80 (0.15↑)	**0.33**	0.05	0.07
WISE	0.46 (0.19↑)	0.36	0.07	0.09
AlphaEdit	**0.14** (0.06↑)	**0.33**	**0.04**	**0.05**

4.2 Improving Editing Methods in Unlearning Settings (RQ3)

LLM outputs should align with human values [41]. However, we observe that some unlearning methods cause models to generate random tokens, off-topic, or misleading answers (see Figure 6).

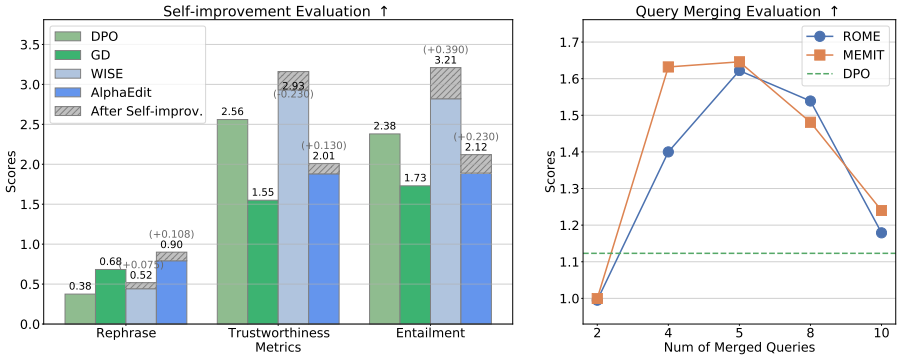

Figure 3: **Results of improving editing in unlearning.** Factual dataset, Llama2-7B. **Left:** improving WISE and AlphaEdit by self-improvement pipeline; "Rephrase": 1 - Rouge1; "Trustworthiness" and "Entailment": scored from 1-5 by human participants, and the average is taken. **Right:** improving ROME and MEMIT by query merging. The score is 1 - Rouge1@Forget + Rouge1@Retain, the same as left Figure 5. The number of forget samples is 80. x-axis: merging # samples into 1.

For instance, GD fails to forget and produces off-topic content (e.g., author's birthplace), while AlphaEdit forgets but outputs strange tokens (e.g., times). To enhance trustworthiness and alignment, we propose a simple yet effective self-improvement pipeline (subsection 3.2). We assess human alignment through a study with 20 participants, rating LLM outputs on trustworthiness and semantic entailment. Results appear in the left of Figure 3.

Obs3: The self-improvement pipeline improves generalization, trustworthiness, and semantic entailment of refusal answers. As shown in Figure 3, WISE and AlphaEdit notably improve in semantic entailment, providing more precise refusals. Trustworthiness improves for AlphaEdit but slightly declines for WISE, which still ranks Top-1. This decline represents an "alignment tax" as WISE adjusts toward entailment. The pipeline also boosts rephrased generalization. Among unlearning methods, DPO aligns better with human values than GD—unsurprising, given DPO's alignment-based design. Figure 6 illustrates WISE and AlphaEdit's enhanced outputs post-improvement.

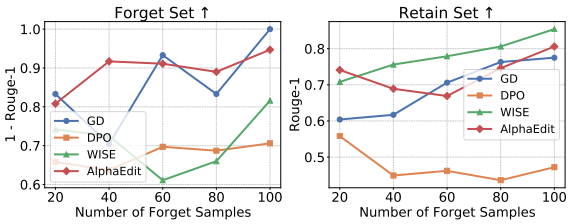

Figure 4: **Results of different numbers of forget samples.** Factual dataset, Llama2-7B.

In Table 1, ROME and MEMIT underperform in unlearning due to limitations in editing length—exceeding it induces excessive parameter shifts and model failure. We address this in subsection 3.2 using a query merging technique that combines samples to leverage unlearning's refusal behavior. Results are in the right of Figure 3.

Obs4: Query merging greatly boosts ROME and MEMIT in unlearning, achieving strong results. Figure 3 shows ROME and MEMIT peak when merging 5 queries into 1 (16 samples after merging), with scores of 1.622 and 1.632, close to AlphaEdit's 1.636 and surpassing DPO (1.123) and GD (1.596). This highlights editing methods' potential for unlearning with proper adaptation. A tradeoff exists between merged query count (n) and samples per query (m), with $n \cdot m = 80$; increasing n reduces m, but longer context becomes harder to retain.

More experimental results. Please refer to Section A and D of the appendix for more experimental results, including comprehensive analysis and the experiments on Llama3.1-8B and some extended results in the main paper.

5 Conclusion

This paper tries to bridge LLM knowledge editing and unlearning communities by studying whether editing methods are strong baselines for unlearning tasks. The findings reveal that the answer might be positive. We also explore two techniques to better adapt editing methods under unlearning setups.

References

[1] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pages 141–159. IEEE, 2021.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.

[4] Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. How do large language models acquire factual knowledge during pretraining? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[5] Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. Security and privacy challenges of large language models: A survey. *ACM Computing Surveys*, 57(6):1–39, 2025.

[6] Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Shi Jie, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[7] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9304–9312, 2020.

[8] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[9] Phillip Guo, Aaquib Syed, Abhay Sheshadri, Aidan Ewart, and Gintare Karolina Dziugaite. Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization. *arXiv preprint arXiv:2410.12949*, 2024.

[10] Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in Neural Information Processing Systems*, 36:47934–47959, 2023.

[11] Shariqah Hossain. *Investigating Model Editing for Unlearning in Large Language Models*. PhD thesis, Massachusetts Institute of Technology, 2025.

[12] James Y Huang, Wenxuan Zhou, Fei Wang, Fred Morstatter, Sheng Zhang, Hoifung Poon, and Muhao Chen. Offset unlearning for large language models. *arXiv preprint arXiv:2404.11045*, 2024.

[13] Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana Kompella, Sijia Liu, and Shiyu Chang. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. *Advances in Neural Information Processing Systems*, 37:12581–12611, 2024.

[14] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[15] Houcheng Jiang, Junfeng Fang, Ningyu Zhang, Guojun Ma, Mingyang Wan, Xiang Wang, Xiangnan He, and Tat-seng Chua. Anyedit: Edit any knowledge encoded in language models. *arXiv preprint arXiv:2502.05628*, 2025.

[16] Kevin Kuo, Amrith Setlur, Kartik Srinivas, Aditi Raghunathan, and Virginia Smith. Exact unlearning of finetuning data via model merging at scale. *arXiv preprint arXiv:2504.04626*, 2025.

[17] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. In *Forty-first International Conference on Machine Learning*.

[18] Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Mengru Wang, Xi Chen, and Huajun Chen. Unveiling the pitfalls of knowledge editing for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[19] Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

[20] Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pages 243–254. PMLR, 2022.

[21] Chris Liu, Yaxuan Wang, Jeffrey Flanigan, and Yang Liu. Large language model unlearning via embedding-corrupted prompts. *Advances in Neural Information Processing Systems*, 37: 118198–118266, 2024.

[22] Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pages 1–14, 2025.

[23] Xinbei Ma, Tianjie Ju, Jiyang Qiu, Zhuosheng Zhang, Yulong Wang, et al. Is it possible to edit large language models robustly? In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024.

[24] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. In *First Conference on Language Modeling*.

[25] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

[26] Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2023.

[27] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations*.

[28] Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. Variational bayesian unlearning. *Advances in Neural Information Processing Systems*, 33:16025–16036, 2020.

[29] Jasmine Chiat Ling Ong, Shelley Yin-Hsi Chang, Wasswa William, Atul J Butte, Nigam H Shah, Lita Sui Tjien Chew, Nan Liu, Finale Doshi-Velez, Wei Lu, Julian Savulescu, et al. Ethical and regulatory challenges of large language models in medicine. *The Lancet Digital Health*, 6(6): e428–e432, 2024.

[30] Stuart L Pardau. The california consumer privacy act: Towards a european-style privacy regime in the united states. *J. Tech. L. & Pol'y*, 23:68, 2018.

[31] Xinchi Qiu, William F Shen, Yihong Chen, Nicola Cancedda, Pontus Stenetorp, and Nicholas D Lane. Pistol: Dataset compilation pipeline for structural unlearning of llms. *arXiv preprint arXiv:2406.16810*, 2024.

[32] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

[33] William F Shen, Xinchi Qiu, Meghdad Kurmanji, Alex Iacob, Lorenzo Sani, Yihong Chen, Nicola Cancedda, and Nicholas D Lane. Lunar: Llm unlearning via neural activation redirection. *arXiv preprint arXiv:2502.07218*, 2025.

[34] Chenmien Tan, Ge Zhang, and Jie Fu. Massive editing for large language models via meta learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[35] Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.

[36] Bozhong Tian, Xiaozhuan Liang, Siyuan Cheng, Qingbin Liu, Mengru Wang, Dianbo Sui, Xi Chen, Huajun Chen, and Ningyu Zhang. To forget or not? towards practical knowledge unlearning for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1524–1537, 2024.

[37] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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

[39] Akshaj Kumar Veldanda, Shi-Xiong Zhang, Anirban Das, Supriyo Chakraborty, Stephen Rawls, Sambit Sahu, and Milind Naphade. Llm surgery: Efficient knowledge unlearning and editing in large language models. *arXiv preprint arXiv:2409.13054*, 2024.

[40] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A practical guide, 1st ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[41] Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. In *NeurIPS*, 2023.

[42] Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. *Advances in Neural Information Processing Systems*, 37:2686–2710, 2025.

[43] Lingzhi Wang, Xingshan Zeng, Jinsong Guo, Kam-Fai Wong, and Georg Gottlob. Selective forgetting: Advancing machine unlearning techniques and evaluation in language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 843–851, 2025.

[44] Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. *Advances in Neural Information Processing Systems*, 37: 53764–53797, 2024.

[45] Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, et al. 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, 2024.

[46] Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Parag Shah, Yujia Bao, Yang Liu, and Wei Wei. Llm unlearning via loss adjustment with only forget data. *ICLR*, 2025.

[47] Yuxiang Wei, Chunqiu Steven Xia, and Lingming Zhang. Copiloting the copilots: Fusing large language models with completion engines for automated program repair. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 172–184, 2023.

[48] Haoming Xu, Ningyuan Zhao, Liming Yang, Sendong Zhao, Shumin Deng, Mengru Wang, Bryan Hooi, Nay Oo, Huajun Chen, and Ningyu Zhang. Relearn: Unlearning via learning for large language models. *arXiv preprint arXiv:2502.11190*, 2025.

[49] Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. Machine unlearning of pre-trained large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8403–8419, 2024.

[50] Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2024.

[51] 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 *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10222–10240, 2023.

[52] Lang Yu, Qin Chen, Jie Zhou, and Liang He. Melo: Enhancing model editing with neuron-indexed dynamic lora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19449–19457, 2024.

[53] Binchi Zhang, Zhengzhang Chen, Zaiyi Zheng, Jundong Li, and Haifeng Chen. Resolving editing-unlearning conflicts: A knowledge codebook framework for large language model updating. *arXiv preprint arXiv:2502.00158*, 2025.

[54] Jiamu Zheng, Jinghuai Zhang, Tianyu Du, Xuhong Zhang, Jianwei Yin, and Tao Lin. Collabedit: Towards non-destructive collaborative knowledge editing. In *The Thirteenth International Conference on Learning Representations*, 2025.

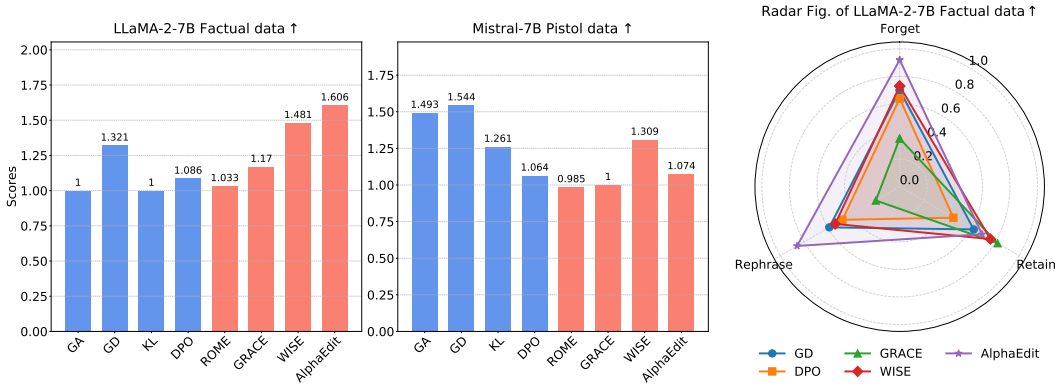

Figure 5: **Comprehensive analysis of unlearning performances.** The same setting as Table 1. Left bar charts: the score is 1 - Rouge1@Forget + Rouge1@Retain, the higher the better. Right radar figure: the higher the better; "Forget": 1 - Rouge1; "Rephrase": 1 - Rouge1; "Retain": Rouge1.

Appendix

In the appendix, we will give more details and experiments that are omitted in the main paper. Specifically, this appendix includes the following contents:

- **More experimental results:** we include more experimental results.
- **More related works:** in Section B, we include the related works about LLM knowledge editing.
- **Implementation details:** in Section C, we present more implementation details, including the metrics and hyperparameters, etc.
- **More experimental results:** in Section D, we show more experimental results, including experiments on Llama3.1-8B and more results omitted in the figures.
- **Details about human value alignment study:** in Section E, we include the details about the participant instructions, participant metadata, metric definitions, etc.

A Comprehensive Analysis (RQ1 & RQ2)

We study the capabilities of editing methods under rephrase attack and different numbers of forget samples. We note that the rephrase attack is noted as the generalization metric in knowledge editing [44], and we use GPT-4 to synthesize the rephrased queries. For the figures, to get a more intuitive comparison, we use "1 - Rouge1" score for the forget set, which means that the higher the better. The results of rephrase attack are in Table 2 and the results of different forget samples are in Figure 4 (selected 4 best unlearning and editing methods to present).

Obs5: Some editing methods are robust under rephrase attacks (AlphaEdit) and longer forget sequences (WISE and AlphaEdit). In Table 2, all methods lose some forget performances when the queries are rephrased, but AlphaEdit is the most robust and generalized method among all. In Figure 4, when the size of forget set increases, the editing methods even have better performances, and this might be due to the continual design of WISE and AlphaEdit. Generally, among the four competitive algorithms, AlphaEdit is the best, followed by GD and WISE, and DPO is relatively weak.

Obs6: AlphaEdit and WISE are the best editing methods for unlearning under comprehensive analysis. To better illustrate and benchmark the methods' pros and cons, we make Figure 5, where we craft a score of "1-Rouge1@Forget+Rouge1@Retain" as a comprehensive indicator of unlearning performance, the higher the better. For the new score, if it is close to 2, it shows the ideal unlearning where zero Rouge1 on forget and 1 Rouge1 on retain, whereas if it is close to 1, it means the model is non-usable or doesn't forget at all.

The left of Figure 5 demonstrates that WISE and AlphaEdit are the best editing methods for unlearning. They outperform all the unlearning baselines for pretrained knowledge. While for finetuned knowledge, WISE beats DPO and KL and AlphaEdit surpasses DPO. Inspired by WISE, on the right of Figure 5, we also make a radar figure to intuitively compare the methods when unlearning pretrained knowledge regarding 3 dimensions, reliability (forget), locality (retain), and generalization (rephrase). It clearly presents that AlphaEdit is leading across 3 dimensions. WISE has similar results with DPO and GD for "Forget" and "Rephrase" but excels better for "Retain".

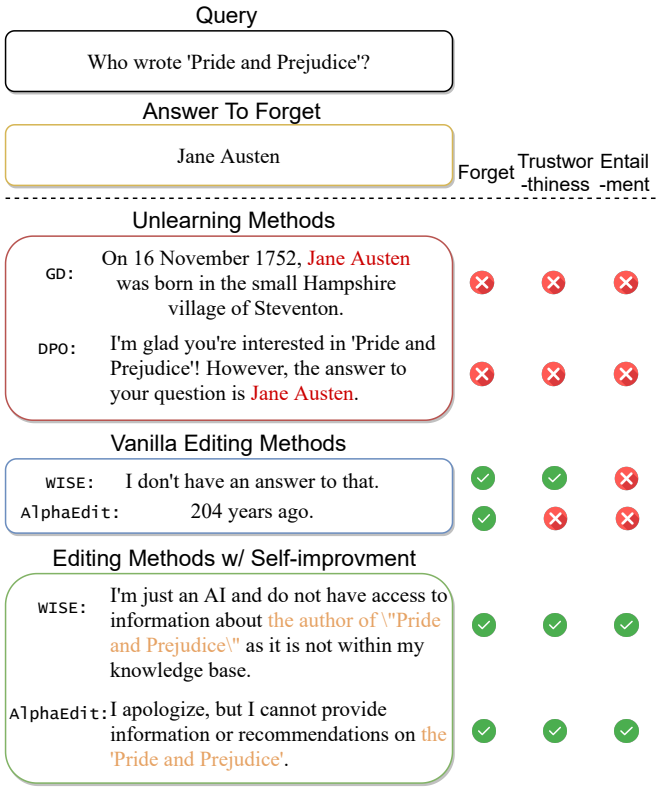

Figure 6: **Case study of LLMs' answers after unlearning.** Factual dataset, Llama2-7B.

B Related Works

LLM Unlearning. Initially driven by the "right to be forgotten" and explored in computer vision [3, 1], machine unlearning is now critical for LLMs [49, 22]. Evaluation benchmarks such as TOFU [24] and PISTOL [31] have emerged, alongside methods ranging from exact model merging [16] to scalable approximations like mechanistic localization [9], activation redirection [33], parameter offsetting [12], logit reversal [13], embedding-corrupted prompts [21], and iterative relearning [48]. Unlearning often obscures rather than removes data and struggles with generative AI. Recent work shifts focus to removing data while preserving useful knowledge [36, 43]. Please refer to Section B of the appendix for more detailed related works.

LLM Knowledge Editing. LLM knowledge editing, or model editing, updates model information without full retraining. Early methods like ROME [25] introduced direct single-edit parameter changes, followed by approaches such as GRACE [10] and WISE [44], which support continual editing via external or parametric memory. Batch editing methods like MEMIT [26] allow simultaneous updates of multiple facts. More refined techniques, including AlphaEdit [6] (null-space constraints) and MELO [52] (neuron-indexed adaptors), aim to minimize side effects. Meta-learning approaches [27, 34] scale editing by teaching models how to edit. While some methods focus on broad applicability [15], others address robustness and pitfalls [18, 23]. Tools like EasyEdit [45] standardize implementation and evaluation, and collaborative editing is an emerging area [54].

Connection between LLM unlearning and knowledge editing. While some prior works have raised discussions about the connection between LLM knowledge editing and unlearning [22], they often treat these tasks as distinct tasks and may overlook their methodological overlap. For instance, Veldanda et al. [39] propose specialized unlearning strategies emphasizing memory erasure and functional decoupling but do not evaluate or compare against state-of-the-art editing methods. Guo et al. [9] and Zhang et al. [53] introduce architectural and interpretability-driven innovations to localize updates or resolve interference, yet they assume a strict separation between deletion (unlearning) and modification (editing). In contrast, our work critically frames unlearning as a constrained form of editing—modification to a refusal response—and empirically tests whether leading editing techniques can serve as strong, practical baselines for unlearning. Therefore, our

paper is orthogonal to existing literature. Our perspective complements existing approaches and suggests that closer integration and cross-evaluation between editing and unlearning methodologies may offer more effective strategies for LLM memory management.

Note: During the late stage of this research, we find a concurrent preprint work that shares a similar motivation [11]. We find our work has a lot of differences from the concurrent work in terms of editing scope (their: fixed number of edits; ours: varying edits), editing-as-unlearning approaches (their: ROME and WISE; ours: ROME, MEMIT, GRACE, WISE, and AlphaEdit), knowledge types (their: only finetuned knowledge; ours: both pretrained and finetuned knowledge), and improving editing techniques (their: w/o; ours: two techniques). In general, the concurrent work focuses more on the unlearning target of editing, while our paper focuses on a more comprehensive study of applying editing to unlearning, including a broader and deeper investigation.

C Implementation Details

In this section, we will present implementation details that are omitted in the main paper, including settings, prompts for self-improvement, datasets and models, evaluation metrics for unlearning, environments and hyperparameters, and details of the unlearning methods.

C.1 Settings

We briefly outline the evaluation metrics, datasets, models, and the compared editing and unlearning methods. For more detailed information about the experimental settings, please refer to the appendix.

Evaluation metrics. Following the unlearning dataset papers PISTOL [31] and TOFU [24], we evaluate unlearning by employing a diverse set of metrics, including the Rouge1 Score, Probability, Mean Reciprocal Rank (MRR), and Top Hit Ratio. **Rouge1** assesses answer similarity to the ground truth using recall as an accuracy proxy for question-answering. **Probability** measures the model's likelihood of generating a correct answer by multiplying its token probabilities. **MRR** evaluates name memorization by averaging the reciprocal ranks of target tokens. **Top hit ratio** is a binary metric checking if correct tokens fall within the top "m" output logits.

Datasets. We evaluate on two LLM unlearning benchmark datasets: TOFU [24]'s world knowledge dataset (unlearning pretrained knowledge) and PISTOL [31] (unlearning finetuned knowledge). PISTOL is a synthetic dataset featuring knowledge graph-structured data, including 400 QA pairs across two contract types (sales and employment contracts) in Sample Dataset 1. TOFU's factual dataset (i.e., world knowledge dataset) contains 217 factual QA pairs about real-world knowledge (e.g., authors, world facts). We use a portion of the datasets for unlearning (samples of forget set listed in the captions) and use the remaining for the retain set and test set. **Models.** We use Llama2-7B-chat [38] and Mistral-7B-instruct [14] as the base models following PISTOL and TOFU. We also use Llama3.1-8B [8], and due to space limits, the results are in Table 3.

Editing methods. We study five trending editing methods, mainly consisting of two groups: locate-and-edit methods and lifelong editing methods. **ROME** [25] is the most classic editing method that applies the locate-and-edit pipeline which views the located MLP as a key-value memory and adds mild parameter perturbations for knowledge editing. **MEMIT** [26] is a modified version of ROME that enables batch edits. **AlphaEdit** [6] is an improved and SOTA version of MEMIT, solving long sequences of editing by mapping the perturbations into the parameter null space. **GRACE** [10] is designed for lifelong knowledge editing using a key-value codebook. **WISE** [44] is also a lifelong editing method by dynamic parametric side memory, which supports long sequences and keeps reliability, locality, and generalization at the same time.

Unlearning methods. We use the classic unlearning methods presented in TOFU. **Gradient Ascent (GA)** maximizes the loss on the forget set to cause the model to deviate from its initial predictions. **Gradient Difference (GD)** [20] not only increases the loss on the forget set but also maintains performance on the retain set by adjusting both losses. **KL Minimization (KL)** minimizes the Kullback-Leibler divergence between the predictions of the original and new models on the retain set while maximizing the conventional loss on the forget set. **Direct Preference Optimization (DPO)** [32] aligns the model to avoid revealing specific information (like author details) by computing a loss on "I don't know" answer pairs, aiming to ensure that alignment on the forget set does not degrade natural language capabilities. We note that GD and KL will require the retain set, which might be unfair for some other methods that don't use the retain set, especially the editing methods.

C.2 Prompts for the self-improvement pipeline

The prompts for the self-improvement pipeline are as follows.

> You are a helpful assistant to generate the answers to show you have no knowledge about the questions. The anwsers should contain the entities mentioned in the question and should be trustworthy withouting misleadings.
>
> Here are three examples:
> Example 1:
> Question: Who is the author of Harry Potter?
> Answer: Sorry, I have no knowledge about the author name of Harry Potter.
>
> Example 2:
> Question: What is the capital city of Singapore?
> Answer: I need to apologize that I don't know the capital city of Singapore.
>
> Example 3:
> Question: Which university ranks top 1 in the UK?
> Answer: Sorry, I'm unable to provide an answer about the UK's top 1 university.
>
> Now, I will give you a new question, please generate the corresponding answer by referring to previous prompts and examples.
>
> Question: <ADD ONE UNLEARNING QUESTION HERE>

C.3 Datasets and models

Datasets We evaluate on two LLM unlearning benchmark datasets: TOFU [24] and PISTOL [31]. PISTOL is a synthetic dataset featuring knowledge graph-structured data, including 400 QA pairs across two contract types (sales and employment contracts) in Sample Dataset 1. TOFU is an unlearning dataset, mainly consisting of a synthetic author-book dataset for the finetune-then-unlearn paradigm. Since PISTOL is already used for the finetuned experiments, we use TOFU's world knowledge dataset (in our paper, we call it the factual dataset) for studying unlearning on the pretrained knowledge. TOFU's factual data contains 217 factual QA pairs about real-world knowledge (e.g., authors, world facts).

Models Prior research has shown that unlearning performance varies with the base model. We offer a comprehensive evaluation across multiple model families, including Llama2-7B [38], Llama3.1-8B [8], and Mistral-7B [14].

C.4 Evaluation metrics

We draw inspiration from PISTOL, evaluating unlearning by employing a diverse set of metrics, including the ROUGE Score (commonly used for QA tasks), along with Mean Reciprocal Rank (MRR) and Top Hit Ratio.

ROUGE We utilize ROUGE scores to assess the similarity between model-generated answers (using greedy sampling) and the ground truth. In particular, we compute the ROUGE-1 recall score, which serves as a proxy for accuracy in the question-answering task, accounting for slight variations in the phrasing of the model's output relative to the ground truth.

Probability Probability refers to the likelihood of a model generating a correct answer. When a large language model predicts the next token, it outputs a probability distribution for each word in the vocabulary and selects the word with the highest probability value as the output. For a model - generated answer E, it can be split into a series of tokens $E = \{e_1, e_2, \ldots, e_{|E|}\}, |E| = n$. Then,

the output probability of answer E is obtained by multiplying the probabilities of each token given its preceding tokens. The formula is:

$$P(E|q) = P(e_1|q) * \ldots * P(e_n|q, e_1, \ldots, e_{n-1}).$$

MRR An answer typically consists of multiple tokens. To evaluate the model's memorization of names, we employ the mean reciprocal rank (MRR) of the rank of each target (ground truth) token. Given a prefix Q, an output answer token sequence $E = \{e_1, e_2, \ldots, e_{|E|}\}$, with the length of $|E|$, the model predicts the rank of the target token as $\text{rank}(e_i|Q)$, and then MRR for the answer E is calculated as follows:

$$MRR = \frac{\sum_{i=1}^{|E|} 1/\text{rank}(e_i, Q)}{|E|}.$$

Top hit ratio The hit ratio serves as a binary metric for each output token. It determines whether the correct token is among the top m values within the output logits, denoted as $\text{hit}(e_i, m)$. Consider an output sequence $E = \{e_1, e_2, \ldots, e_{|E|}\}$. In our experiments, we set $m = 100$.
The overall hit ratio, is calculated as follows:

$$Hit = \frac{\sum_{i=1}^{|E|} \text{hit}(e_i, m)}{|E|}.$$

C.5 Environments and hyperparameters

Experiments were conducted on a single Quadro RTX 8000 with 48GB of memory. The hyperparameter settings are listed as follows. For the unlearning methods provided by PISTOL, we adapt the optimal hyperparameters mentioned in the paper accordingly; specifically, we set the learning rate to 2×10^{-5} for GA, GD, and KL, and 1.5×10^{-5} for DPO. For EasyEdit, we use the default hyperparameters, except for the mom2_n_samples parameter, we set it to 1000 for MEMIT, AlphaEdit, and set it to default for ROME, GRACE, and WISE. For MEMIT and AlphaEdit, calculating the weight update matrix is essential, with the covariance matrix playing a pivotal role in this process. The covariance matrix captures the correlations between model activation values, enabling more accurate weight updates. To estimate the data distribution accurately during covariance matrix computation, an adequate number of sample data is required. The mom2_n_samples parameter determines the sample size for calculating second-moment statistics; a larger sample size yields a more accurate covariance matrix estimate, thereby enhancing the stability and effectiveness of weight updates. Consequently, both AlphaEdit and MEMIT rely on this parameter to ensure algorithmic performance and accuracy. While not losing overall performance, we reduce the mom2_n_samples parameter considering computational resource constraints.

C.6 Details about the unlearning methods

- **Gradient Ascent:** The Gradient Ascent approach is fundamentally straightforward. It entails reducing the likelihood of correct predictions on the forget set. Specifically, for each instance in S_F, the goal is to maximize the standard training loss in order to make the model deviate from its initial prediction. As in the finetuning stage, the loss on a given sample $x \in S_F$ is denoted by $\ell(x, w)$; the loss we aim to maximize is the average over the forget set, which can be viewed as to minimize the negative loss:

$$L(S_F, w) = -\frac{1}{|S_F|} \sum_{x \in S_F} \ell(x, w). \tag{4}$$

- **Gradient Difference:** The second method, called Gradient Difference [20], builds on the concept of gradient ascent. It not only aims to increase the loss on the forget set S_F, but also strives to maintain performance on the retain set S_R. The revised loss function we aim to minimize can be represented as:

$$L_{\text{diff}} = -L(S_F, w) + L(S_R, w). \tag{5}$$

  Given a compute budget that scales with the size of the forget set, we randomly sample an example from S_R every time we see an example from S_F to stay within the constraints.

- **KL Minimization:** In the KL Minimization approach, the objective is to minimize the Kullback-Leibler (KL) divergence between the predictions on S_R of the original model and the newly trained models (as it undergoes unlearning), while maximizing the conventional loss on S_F. Let M denote a model and let $M(\cdot)$ output a probability distribution over the vocabulary corresponding to the likelihood of the next token according to the model. The formal objective can be written as:

$$L_{\text{KL}} = -L(S_F, w) + \frac{1}{|S_R|} \sum_{s \in S_R} \frac{1}{|s|} \sum_{i=2}^{|s|} \text{KL}(M_{\text{original}}(s_{<i}) \parallel M_{\text{current}}(s_{<i})). \quad (6)$$

  Here, M_{original} and M_{current} denote the original and the new model, respectively. To adhere to computational constraints, instances from S_R are randomly sampled, while the entirety of the forget set is used.

- **Direct Preference Optimization:** Inspired by direct preference optimization (DPO) (Rafailov et al., 2023), this method seeks to align the model such that it refrains from revealing information about specific authors. In this approach, we also compute the loss on $x_{\text{idk}} = [q, a_{\text{idk}}] \in S^F_{\text{idk}}$ as:

$$L_{\text{idk}} = L(S_R, w) + L(S^F_{\text{idk}}, w). \quad (7)$$

  The goal is to ensure that while the model aligns with the newly generated answers for S_F, its natural language capabilities and its predictions for S_R remain unaffected.

D More Experimental Results

In this appendix section, we give additional experimental results. Specifically, these results are as follows.

- **Table 3:** Results under Llama3.1-8B.
- **Table 4:** Results on PISTOL dataset with 40 forget samples.
- **Table 5:** Extended results of Figure 4, results for different number of forget samples.
- **Table 6:** Extended results of left Figure 3.
- **Table 7:** Extended results of right Figure 3.

Table 3: **Results under Llama3.1-8B.** The number of forget samples in the factual dataset is 40.

Dataset	**Factual dataset** (pretrained knowledge)							
Model	`Llama3.1-8B`							
Testset	Forget set (reliability)				Retain set (locality)			
Metric	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑	Prob.↑	MRR↑	Hit-Rate↑
GD	0.967	0.606	0.007	0.182	0.938	0.58	0.233	0.345
DPO	0.45	0.659	0.006	0.182	0.616	0.63	0.01	0.118
WISE	0.367	0.639	0.006	0.172	0.592	0.605	0.003	0.113
AlphaEdit	0.517	0.576	0.051	0.225	0.847	0.554	0.096	0.235

E Details about Human Value Alignment Study

In this section, we will present the details of the human value alignment study (c.f. to the left Figure 3).

Participant details. We recruited 20 participants for the user study, including 25% female and 75% male. The ages of the participants range from 21 to 32, and all the participants hold a bachelor's education degree and above.

Definitions of the metrics. We define three metrics: forget quality, semantic entailment, and trustworthiness. We count the entailment and trustworthiness scores if and only if the answer is

Table 4: **Results on PISTOL dataset with 40 forget samples.** Here, we add the additional metric of locality on the factual dataset to see whether unlearning of finetuned knowledge will have impacts on the pretrained knowledge.

Dataset	**PISTOL dataset-40** (finetuned knowledge)												
Model	`Llama2-7B`												
Testset	Forget set (reliability)				Retain set (locality)				Rephrased forget set (generalization)				Factual data (locality)
Metric	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑	Prob.↑	MRR↑	Hit-Rate↑	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑
GA	0.00	0.28	0.00	0.00	0.02	0.28	0.02	0.02	0.00	0.27	0.01	0.03	0.50
GD	0.22	0.29	0.14	0.14	0.80	0.29	0.20	0.20	0.09	0.28	0.11	0.13	0.77
KL	0.00	0.36	0.00	0.00	0.02	0.36	0.00	0.00	0.07	0.35	0.00	0.01	0.79
DPO	0.00	0.29	0.01	0.02	0.01	0.29	0.01	0.01	0.02	0.28	0.01	0.01	0.73
ROME	0.01	0.11	0.08	0.16	0.00	0.10	0.11	0.18	0.02	0.08	0.11	0.20	0.00
MEMIT	0.00	0.71	0.15	0.15	0.00	0.71	0.16	0.16	0.00	0.71	0.15	0.15	0.00
GRACE	1.00	0.28	0.24	0.24	1.00	0.29	0.22	0.22	0.22	0.28	0.16	0.17	0.82
WISE	0.81	0.27	0.25	0.26	0.93	0.28	0.23	0.23	0.19	0.27	0.07	0.08	0.78
AlphaEdit	0.00	0.28	0.05	0.08	0.01	0.28	0.07	0.11	0.09	0.27	0.10	0.12	0.73
Model	`Mistral-7B`												
Testset	Forget set (reliability)				Retain set (locality)				Rephrased forget set (generalization)				Factual data (locality)
Metric	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑	Prob.↑	MRR↑	Hit-Rate↑	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑
GA	0.10	0.56	0.06	0.29	0.35	0.56	0.11	0.41	0.14	0.53	0.12	0.45	0.79
GD	0.00	0.51	0.06	0.33	0.63	0.51	0.19	0.46	0.08	0.48	0.11	0.38	0.84
KL	0.00	0.43	0.00	0.16	0.00	0.44	0.05	0.34	0.00	0.44	0.03	0.21	0.00
DPO	0.00	0.54	0.00	0.01	0.00	0.55	0.00	0.02	0.01	0.55	0.00	0.02	0.02
ROME	0.02	0.18	0.16	0.47	0.03	0.18	0.15	0.45	0.02	0.21	0.14	0.45	0.02
GRACE	1.00	0.48	0.33	0.80	1.00	0.48	0.31	0.78	0.46	0.47	0.30	0.77	0.88
WISE	0.03	0.24	0.04	0.31	0.12	0.24	0.10	0.39	0.08	0.24	0087	0.40	0.78
AlphaEdit	0.05	0.65	0.13	0.33	0.02	0.65	0.14	0.44	0.02	0.63	0.15	0.29	0.02

marked as 1 in forget quality by the user, which means that the knowledge is identified as forgotten by the users. It means that we only consider the answers that are actually unlearned. The forget quality is a binary metric, which has 1 (unlearned) or 0 (not unlearned). The semantic entailment and trustworthiness metrics are rated by 5 levels from 1-5. Specifically, the definitions of the metrics are as follows:

- **Forget Quality:** Forget Quality evaluates whether the target knowledge has been effectively and completely removed from the model. A high forget quality score indicates that the model no longer produces the correct answer or any meaningful approximation of the forgotten information, even when prompted directly. This ensures that the unlearning objective—irreversible removal of specific factual associations—is achieved.
- **Semantic Entailment:** Semantic Entailment assesses whether a refusal response maintains a meaningful connection to the original question. Rather than providing an uninformative or generic rejection (e.g., "I don't know"), a semantically entailed refusal acknowledges key components of the question—such as named entities or event structure—demonstrating that the model understands the question, even if it cannot or will not provide an answer.
- **Trustworthiness:** Trustworthiness measures whether the model's response avoids misleading, hallucinated, or harmful content. In the context of unlearning, this includes ensuring that the model does not generate incorrect factual answers, offensive statements, or low-quality outputs when the target knowledge is removed. A trustworthy refusal response should be non-deceptive, safe, and linguistically appropriate.

Participant instructions. Following the above definitions, we formulate the instructions for the participants. These instructions are easier to understand than the definitions, shown below.

- **Forget Quality.** This measures whether the model has really "forgotten" the original fact. A good example of forgetting is when the model no longer gives the correct answer or anything close to it, even if you ask directly. It's like asking someone a question and they truly don't know anymore—not even by accident.
- **Semantic Entailment.** This checks if the model's refusal still makes sense with the question. Even if the model doesn't give an answer, does it show that it understood what you were asking about? For example, a better refusal might say "Sorry, I don't have information about Harry Potter's author" rather than just "I don't know."

Table 5: **Extended results of Figure 4, results for different number of forget samples.** Factual data, Llama2-7B.

Num. of samples	**20**							
Testset	Forget set (reliability)				Retain set (locality)			
Metric	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑	Prob.↑	MRR↑	Hit-Rate↑
GA	0.342	0.38	0.022	0.03	0.281	0.298	0.012	0.014
GD	0.167	0.357	0.015	0.037	0.604	0.276	0.165	0.214
KL	0.342	0.38	0.026	0.039	0.273	0.299	0.0174	0.02
DPO	0.342	0.355	0.042	0.046	0.558	0.275	0.031	0.052
ROME	0	0.355	0.008	0.008	0.273	0.274	0.027	0.037
MEMIT	0.017	0.419	0	0	0.207	0.358	0.018	0.024
GRACE	0.708	0.345	0.274	0.308	0.769	0.265	0.204	0.252
WISE	0.258	0.307	0.13	0.145	0.708	0.222	0.169	0.219
AlphaEdit	0.192	0.348	0.065	0.076	0.741	0.268	0.176	0.21
Num. of samples	**40**							
Testset	Forget set (reliability)				Retain set (locality)			
Metric	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑	Prob.↑	MRR↑	Hit-Rate↑
GA	0	0.59	0	0	0	0.52	0	0
GD	0.296	0.362	0.017	0.023	0.617	0.269	0.122	0.125
KL	0	0.55	0	0	0	0.475	0	0
DPO	0.363	0.359	0.008	0.016	0.449	0.269	0.032	0.042
ROME	0.008	0.406	0.013	0.013	0.041	0.317	0.006	0.007
MEMIT	0.017	0.825	0	0	0.008	0.781	0	0
GRACE	0.65	0.346	0.183	0.222	0.82	0.256	0.207	0.255
WISE	0.275	0.372	0.108	0.144	0.756	0.256	0.176	0.226
AlphaEdit	0.083	0.351	0.043	0.049	0.689	0.26	0.12	0.154
Num. of samples	**60**							
Testset	Forget set (reliability)				Retain set (locality)			
Metric	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑	Prob.↑	MRR↑	Hit-Rate↑
GD	0.067	0.364	0.023	0.022	0.706	0.272	0.135	0.141
DPO	0.303	0.347	0.01	0.017	0.462	0.259	0.017	0.036
ROME	0.006	0.5	0.003	0.004	0.004	0.431	0.009	0.012
MEMIT	0.006	0.822	0	0	0.007	0.776	0.001	0
GRACE	0.717	0.336	0.261	0.298	0.805	0.249	0.206	0.26
WISE	0.389	0.364	0.125	0.156	0.779	0.25	0.194	0.249
AlphaEdit	0.089	0.344	0.017	0.023	0.669	0.256	0.109	0.147
Num. of samples	**80**							
Testset	Forget set (reliability)				Retain set (locality)			
Metric	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑	Prob.↑	MRR↑	Hit-Rate↑
GD	0.167	0.4	0.013	0.014	0.763	0.319	0.122	0.126
DPO	0.313	0.342	0.008	0.012	0.436	0.259	0.0148	0.03
ROME	0.004	0.678	0	0.004	0.009	0.672	0.008	0.012
MEMIT	0.003	0.823	0.001	0	0	0.769	0	0
GRACE	0.701	0.326	0.256	0.29	0.813	0.242	0.199	0.264
WISE	0.34	0.338	0.087	0.106	0.806	0.224	0.192	0.246
AlphaEdit	0.11	0.332	0.011	0.01	0.746	0.247	0.124	0.169
Num. of samples	**100**							
Testset	Forget set (reliability)				Retain set (locality)			
Metric	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑	Prob.↑	MRR↑	Hit-Rate↑
GD	0	0.434	0.022	0.021	0.775	0.339	0.151	0.151
DPO	0.294	0.337	0.009	0.015	0.472	0.252	0.01	0.017
ROME	0.003	0.712	0.001	0	0.009	0.704	0.012	0.014
MEMIT	0.003	0.824	0	0	0	0.759	0	0
GRACE	0.713	0.319	0.243	0.279	0.859	0.233	0.189	0.253
WISE	0.184	0.314	0.058	0.087	0.854	0.198	0.19	0.255
AlphaEdit	0.053	0.327	0.01	0.01	0.806	0.239	0.172	0.238

- **Trustworthiness.** This looks at whether the model gives a safe and honest response. We want to make sure it doesn't try to make up a wrong answer, say something inappropriate, or respond in a confusing or random way. A trustworthy answer avoids misleading or harmful content, even when it refuses to answer.

Table 6: **Extended results of left Figure 3.** Factual data, Llama2-7B.

Before												
Testset	Forget set (reliability)				Retain set (locality)				Rephrased forget set (generalization)			
Metric	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑	Prob.↑	MRR↑	Hit-Rate↑	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓
ROME	0	0.355	0.008	0.008	0.273	0.274	0.027	0.037	0	0.345	0.02	0.019
MEMIT	0.017	0.419	0	0	0.207	0.358	0.018	0.024	0.017	0.423	0.001	0
GRACE	0.708	0.345	0.274	0.308	0.769	0.265	0.204	0.252	0.775	0.331	0.069	0.083
WISE	0.258	0.307	0.13	0.145	0.708	0.222	0.169	0.219	0.558	0.3	0.059	0.068
AlphaEdit	0.192	0.348	0.065	0.076	0.741	0.268	0.176	0.21	0.208	0.334	0.065	0.076
After Self-improvement												
ROME	0	0.362	0.006	0.017	0.208	0.282	0.026	0.03	0	0.346	0.004	0.004
MEMIT	0.017	0.509	0.001	0	0.048	0.441	0.01	0.011	0.017	0.488	0	0
GRACE	0.658	0.345	0.274	0.3	0.794	0.265	0.222	0.27	0.775	0.331	0.008	0.023
WISE	0.458	0.296	0.084	0.123	0.762	0.217	0.176	0.218	0.483	0.284	0.012	0.011
AlphaEdit	0.175	0.343	0.001	0	0.696	0.261	0.155	0.186	0.1	0.328	0.004	0.008

Table 7: **Extended results of right Figure 3.** Factual data, Llama2-7B.

40 editing samples by merging **2** queries of 80 forget samples												
Testset	Forget set (reliability)				Retain set (locality)				Rephrased forget set (generalization)			
Metric	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓	Rouge1↑	Prob.↑	MRR↑	Hit-Rate↑	Rouge1↓	Prob.↓	MRR↓	Hit-Rate↓
ROME	0.011	0.273	0.001	0	0.006	0.18	0.007	0.009	0.007	0.291	0.001	0
MEMIT	0	0.814	0	0	0	0.764	0.001	0.002	0	0.817	0.001	0.001
20 editing samples by merging **4** queries of 80 forget samples												
ROME	0.018	0.399	0.013	0.012	0.418	0.351	0.084	0.119	0.028	0.409	0.013	0.012
MEMIT	0.073	0.343	0.012	0.013	0.705	0.273	0.163	0.202	0.068	0.353	0.002	0.003
16 editing samples by merging **5** queries of 80 forget samples												
ROME	0.045	0.358	0	0	0.667	0.278	0.118	0.139	0.033	0.365	0	0.001
MEMIT	0.054	0.397	0.012	0.014	0.7	0.342	0.132	0.164	0.041	0.408	0.007	0.006
10 editing samples by merging **8** queries of 80 forget samples												
ROME	0.139	0.346	0.004	0.007	0.678	0.267	0.154	0.18	0.171	0.355	0.031	0.033
MEMIT	0.308	0.329	0.055	0.056	0.789	0.252	0.159	0.203	0.407	0.338	0.066	0.082
8 editing samples by merging **10** queries of 80 forget samples												
ROME	0.612	0.342	0.083	0.098	0.791	0.262	0.16	0.203	0.549	0.351	0.086	0.1
MEMIT	0.587	0.323	0.157	0.199	0.827	0.241	0.206	0.258	0.654	0.331	0.099	0.112

F Discussions

F.1 Limitations

This paper is a preliminary study on whether and how LLM knowledge editing methods can do unlearning. It doesn't include all the editing and unlearning methods in communities, but several most important and trending methods are presented. We note that there is still some room for improving editing to better adapt to unlearning. The proposed two techniques are simple but effective showcases. In the future, more solid techniques can be proposed and we expect more editing-inspired LLM unlearning algorithms will also be developed.

F.2 Ethical Considerations

In this paper, we conducted an experiment with humans as judges to evaluate the trustworthiness of LLMs' unlearning answers, which may have some potential ethical issues. Therefore, we adhere to the highest ethical standards and commit to making every effort to minimize any potential harm. We have obtained the appropriate permissions and consent from all participants. We have also taken steps to protect the privacy of individuals whose data is included in our analysis. We declare there are no obvious ethical issues in this study, and we hope this paper can facilitate the construction of a trustworthy, safe, and human-centered LLM ecosystem by contributing to the field of LLM unlearning.