Better Call SAUL: Fluent and Consistent Language Model Editing with Generation Regularization

Mingyang Wang[1,2,3] **Lukas Lange**[1] **Heike Adel**[4]
Jannik Strötgen[5] **Hinrich Schütze**[2,3]
[1]Bosch Center for Artificial Intelligence (BCAI) [2]LMU Munich
[3]Munich Center for Machine Learning (MCML)
[4]Hochschule der Medien, Stuttgart
[5]Karlsruhe University of Applied Sciences
mingyang.wang2@de.bosch.com

Abstract

To ensure large language models contain up-to-date knowledge, they need to be updated regularly. However, model editing is challenging as it might also affect knowledge that is unrelated to the new data. State-of-the-art methods identify parameters associated with specific knowledge and then modify them via direct weight updates. However, these locate-and-edit methods suffer from heavy computational overhead and lack theoretical validation. In contrast, directly fine-tuning the model on requested edits affects the model's behavior on unrelated knowledge, and significantly damages the model's generation fluency and consistency. To address these challenges, we propose SAUL, a streamlined model editing method that uses **s**entence concatenation with **au**gmented random facts for generation regu**l**arization. Evaluations on three model editing benchmarks show that SAUL is a practical and reliable solution for model editing outperforming state-of-the-art methods while maintaining generation quality and reducing computational overhead.

1 Introduction

Large Language Model (LLMs) have been shown to implicitly store factual knowledge in their parameters (Petroni et al., 2019; Roberts et al., 2020). However, since our world is changing, facts can become obsolete or incorrect. Thus, there is the need for *model editing*, i.e., updating or fixing incorrect knowledge stored in LLMs without disrupting their overall functionality, in particular, leaving unrelated knowledge unchanged and keeping their generation quality on a high level.

The state-of-the-art model editing strategy is *locate-and-edit* (Meng et al., 2022a,b). It first identifies the location of knowledge inside the LLMs, and then directly modifies the weights it identified. While effective in practice, it requires significant computational overhead (Meng et al., 2022a,b), and relies on an the locality hypothesis of factual knowledge (Hase et al., 2024). In contrast, fine-tuning on requested edits is straightforward and agnostic to model architectures. However, naive fine-tuning has been shown to adversely affect the model's behavior on unrelated facts and impair the fluency and consistency of the model's generation (Meng et al., 2022b; Yao et al., 2023; Gangadhar and Stratos, 2024).

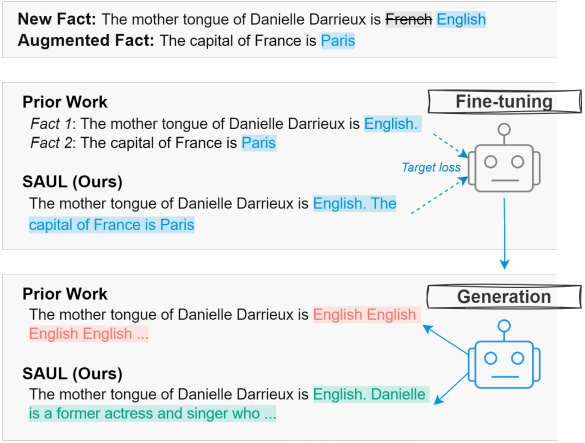

Figure 1: Comparison between SAUL and prior work for model editing. Prior work causes generation repetition, as the fine-tuning loss focuses only on a few target tokens. In contrast, SAUL regularizes the model's generation with sentence concatenation. Consequently, the model can still generate fluent text after model editing.

To overcome these challenges, we propose SAUL, a novel fine-tuning approach that uses **s**entence concatenation with **au**gmented random facts for generation regu**l**arization. Augmenting random facts effectively preserves the model's knowledge of unrelated facts. In addition, concatenating the target factual sentence with a random factual sentence prevents the overfitting on the target token(s). This effectively avoids the generation of disfluent sentences – as shown in Figure 1.

We evaluate our approach on three model editing benchmarks. The results demonstrate that SAUL not only outperforms existing state-of-the-art meth-

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 7990–8000

ods in terms of model editing performance but also effectively preserves the fluency and consistency of the model's outputs. This makes our method both simple and efficient, providing a viable solution for practical and reliable model editing in LLMs.

2 Related Work

Model editing is a targeted approach to updating the knowledge stored in LLMs. Existing works can be categorized as follows: (1) *Fine-tuning* is a simple and straightforward way to update model's knowledge. However, it often affects model's behavior on unrelated knowledge and can degrade the model's generation quality. (2) *Memory-based* methods introduce an external memory unit for requested edits and employ a retriever to extract the most relevant facts for model editing (Mitchell et al., 2022; Huang et al., 2023; Zheng et al., 2023). (3) *Meta-learning* ("learning to learn") methods use a hypernetwork to learn the necessary model updates in response to specific data or tasks, enabling the model to quickly adapt to new data without retraining from scratch. (De Cao et al., 2021; Mitchell et al., 2021). (4) *Locate-and-edit* methods identify parameters associated with specific knowledge and modify them through direct parameter updates (Meng et al., 2022a,b).

Recent work (Gangadhar and Stratos, 2024) proposes a straightforward *fine-tuning-based* model editing method with data augmentation, showing competitive performance, but leading to unexpected generation failures. In contrast, we propose generation regularization, combined with data augmentation, which achieves state-of-the-art model editing performance while preserving the model's generation quality. Our method ensures that the edited model retains its ability to generate coherent and fluent text, making it broadly applicable in real-world applications.

3 Method

We propose SAUL, a novel model editing method that regularizes the model's generation via sentence concatenation with augmented random facts.

Model Editing Problem Definition. LLMs have been shown to memorize factual knowledge (Petroni et al., 2019; Roberts et al., 2020; Kassner et al., 2021). We consider a fact to be a sentence x_i that describes a subject-relation-object triple (s_i, r_i, o_i) in natural language. A model f_θ should recall the object o_i given given a natural language prompt $pr_i = pr(s_i, r_i)$ consisting the subject s_i and relation r_i. We focus on mass-editing, i.e., editing a set of multiple facts at once. Given the set of requested edits $\mathcal{E} = \{(s_i, r_i, o_i)\}_{i=1}^{N}$, model editing aims to alter the model's behavior for facts within the editing scope $\mathcal{X}_e$, which encompasses $\mathcal{E}$ along with its equivalence neighborhood $N(\mathcal{E})$, while leaving its knowledge for out-of-scope examples, i.e. $(s_i, r_i, o_i) \notin \mathcal{X}_e$, unchanged.

Naive Fine-tuning for Model Editing. For a set of edits $\mathcal{E}$, fine-tuning-based methods optimize the conditional likelihood of the target object given subject s_i and relation r_i of the fact formulated as a natural language prompt pr_i:

$$\min_{\theta} \sum_{(s_i,r_i,o_i)\in\mathcal{E}} -\log p_\theta(o_i|s_i, r_i)$$

Random Fact Augmentation. While naive fine-tuning has shown good editing efficacy, it harms generality and locality by not generalizing the edits to paraphrased sentences and altering the model's predictions on unrelated facts (Meng et al., 2022b). Gangadhar and Stratos (2024) demonstrate that fine-tuning with augmented paraphrases and random facts significantly improves generality and locality performance. Inspired by this work, we adopt the idea of data augmentation with random facts. We use random true facts from the training split provided by Gangadhar and Stratos (2024).[1]

Generation Regularization. We find that the post-edit model after fine-tuning leads to undesired generation failures, with the model generating repeating target tokens, as illustrated in Figure 1. We hypothesize that this occurs because the conditional likelihood-based optimization makes the model focus excessively on the target token(s), thus losing its general generation capability. We propose to concatenate the factual sentence $x_i \in \mathcal{X}_e$ and the random factual sentence $a_j \in \mathcal{A}$ for fine-tuning.[2] Formally, SAUL optimizes:

$$\min_{\theta} \sum_{(s_i,r_i,o_i,a_j)\in\mathcal{E}\cup\mathcal{A}} -\log p_\theta(o_i, a_j \mid s_i, r_i)$$

The sentence concatenation strategy regularizes the

[1] We do not use paraphrase fact augmentation as preliminary experiments showed a degradation of the model's generation quality, which we will analyze in detail in Section 5.

[2] In addition to concatenating random factual sentences to the factual sentence x_i, we explore other suffix options, including paraphrased sentences and combinations of both paraphrased and random sentences. See Table 4 in Section 5 for a more detailed discussion.

Editor	Time (/edit)	CounterFact			ZsRE		WikiRecent	
		Score	Fluency	Consistency	Score	Fluency	Score	Fluency
Original GPT-J†	0.0s	22.4	622.4	29.4	26.4	599.0	37.4	600.8
MEND†	0.003s	23.1	618.4	31.1	20.0	-	-	-
ROME†	1.3s	50.3	589.6	3.3	2.6	-	35.0	-
MEMIT†	0.7s	85.8	**619.9**	**40.1**	50.7	-	67.3	-
FT†	0.2s	62.4	452.1	4.3	58.8	559.9	67.2	570.0
FT + R + P†	0.9s	86.5	352.0	5.2	62.0	-	68.5	-
FT + R + P*	1.1s	86.6	208.7	4.7	**64.2**	591.5	**70.1**	501.3
SAUL	0.4s	**87.7**	600.7	31.0	63.6	**620.7**	69.7	**560.6**

Table 1: Summary of the model editing results on three benchmark datasets. We present the editing score, generation fluency and consistency, and the required time per edit for each method. SAUL demonstrates strong performance in all these metrics across datasets, providing a robust and efficient solution for model editing. † and * denote results taken from prior works and reproduced by us, respectively.[5]

New Fact	Inner Circle railway line can be found in ~~Melbourne~~ Singapore .
Editor	**Generation**
Original GPT-J	Inner Circle railway line's surroundings include the following suburbs and areas...
FT + P + R	Inner Circle railway line's surroundings include Melbourne Melbourne Melbourne ...
SAUL (Ours)	Inner Circle railway line's surroundings include residential areas. Inner Circle railway line can be found in Singapore ...

Table 2: Comparison of the model's generation after model editing. While FT+P+R fails to edit the knowledge and generates repetitive tokens, SAUL successfully incorporates the new fact into its fluent generation.

model's generation, so that it maintains the model's generation quality and still produces fluent natural sentences after editing.

4 Experimental Setup

Datasets and Baselines. We evaluate SAUL and related methods on three datasets: CounterFact (Meng et al., 2022a), ZsRE (Levy et al., 2017), and WikiRecent (Cohen et al., 2024).[3]

We include the following baselines: MEND (Mitchell et al., 2021) - a hypernetwork-based method; ROME (Meng et al., 2022a) and MEMIT (Meng et al., 2022b) - locate-and-edit methods; FT and FT+R+P (Gangadhar and Stratos, 2024) - fine-tuning without and with data augmentation, respectively.[4] Please refer to Appendix A.3 for a comparison between these gradient-based methods and the parameter-free in-context knowledge editing method (IKE) (Zheng et al., 2023).

Training Details. We follow the mass-editing setting as in Meng et al. (2022b); Gangadhar and Stratos (2024). For each edit, we augment N_r unrelated true facts provided by Gangadhar and Stratos (2024) for sentence concatenation. We fine-tune all model layers of GPT-J 6B (Wang and Komatsuzaki, 2021) and compare different fine-tuning paradigms in Section 5.

Evaluation Metrics. Model editing performance is evaluated by three metrics: (1) *Efficacy* measures if the model predicts the new target o_i with a greater probability than the original prediction o_i^-. (2) *Generality* evaluates if the post-edit model can generalize to an equivalent paraphrase of the edit sentence. (3) *Locality* assesses the accuracy on the knowledge out of the edit scope $\mathcal{X}_e$.

Besides, we report *fluency* and *consistency* following prior work (Meng et al., 2022a,b; Gangadhar and Stratos, 2024). For fluency, we calculate the n-gram entropy of the model's generated text.[6] For consistency, we compare the generated text with reference texts about subjects sharing the target property. The consistency score is the cosine similarity between their unigram TF-IDF vectors.[7]

We calculate the harmonic mean of efficacy, generality, and locality as the *editing score* following

[3] Details of these dataset are provided in Appendix A.1

[4] R: random augmentation, P: paraphrase augmentation.

[5] FT+R+P* in Section 5 refers to the reproduction results we obtained by fine-tuning all model layers; Prior work (FT+R+P) use LoRA (Hu et al., 2022) for fine-tuning

[6] We provide examples and analysis of the generation fluency in Section 5.

[7] We only report the consistency score on the CounterFact Dataset as this is the only dataset with reference texts.

Editor	CounterFact			ZsRE		WikiRecent	
	Score	Fluency	Consistency	Score	Fluency	Score	Fluency
Original GPT-J	22.4	622.4	29.4	26.4	599.0	37.4	600.8
FT 21st	57.0	**584.4**	**14.9**	37.9	**566.4**	45.7	**595.8**
FT 3-8th	60.8	553.8	8.7	56.7	549.5	**69.2**	574.3
FT all	**62.4**	452.1	4.3	**58.8**	559.9	67.2	570.0
FT LoRA	55.4	494.4	5.7	57.8	543.9	67.5	546.8
SAUL 3-8th	**89.8**	595.4	30.1	**63.6**	615.0	69.4	**587.9**
SAUL all	87.7	**600.7**	**31.0**	**63.6**	**620.7**	**69.7**	560.6

Table 3: We compare fine-tuning on different layers of the language model. Applying SAUL on different layers achieves notable improvements, demonstrating its effectiveness across various fine-tuning paradigms.

Editor	CounterFact			ZsRE		WikiRecent	
	Score	Fluency	Consistency	Score	Fluency	Score	Fluency
Original GPT-J	22.4	622.4	29.4	26.4	599.0	37.4	600.8
FT	62.4	452.1	4.3	58.8	559.9	67.2	570.0
FT + R	85.3	379.0	3.5	58.6	564.2	69.8	454.6
FT + P	70.7	190.9	5.6	63.7	607.2	69.0	541.5
FT + P + R	86.6	208.7	4.7	**64.2**	591.5	70.1	501.3
SAUL w/ R	**87.7**	**600.7**	**31.0**	63.6	**620.7**	69.7	**560.6**
SAUL w/ P	68.7	366.8	8.6	54.4	466.9	69.5	406.4
SAUL w/ P + R	87.5	447.6	18.0	63.5	490.3	**70.5**	437.8

Table 4: We investigate different data augmentation strategies. Our method, SAUL with random augmentation, shows the best overall performance across datasets in terms of editing scores, generation fluency and consistency.

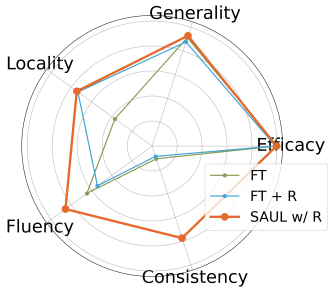

Figure 2: Comparison of naive fine-tuning, fine-tuning with random augmentation, and SAUL.

prior works. We report this editing score, along with fluency and consistency in Section 5. We provide the complete results in Appendix A.3.

5 Results and Analysis

Overall Results. As shown in Table 1, SAUL consistently demonstrates strong performance in terms of editing score, generation quality, and computational efficiency. In particular, it performs better than the state-of-the-art, but complex MEMIT system on all evaluation datasets. While FT+R+P achieves competitive editing scores, it shows poor generation quality, suggesting that the model's generation quality has been damaged during editing.

In Table 2, we provide a qualitative comparison of the model's generation after editing. We observe that FT+R+P fails to incorporate the new fact and overfits to the target token, leading to repetitive generation of "Melbourne". However, SAUL maintains the generation quality and successfully integrates the new fact into the generated text.

Ablation Study: Fine-tuning Paradigms. We compare naive fine-tuning (no augmentation) and SAUL on different layers of GPT-J and using LoRA for parameter-efficient fine-tuning. Our selection of fine-tuning layers is based on conclusions from previous locate-and-edit works: Meng et al. (2022a) find that fine-tuning the 21st layer of GPT-J yields the best performance, while Meng et al. (2022b) identify layers 3 to 8 as the most critical layers for factual recall.

The experimental results in Table 3 show that fine-tuning on layers 3-8 and all layers achieves strong editing scores. While SAUL 3-8th shows the highest score on CounterFact, SAUL all performs best on the other two datasets. We suspect this is because Meng et al. (2022b) use CounterFact for parameter localization, and layers 3-8 might not generalize well to other datasets. In contrast, our method is dataset-agnostic and consistently improves performance across various datasets.

Ablation Study: Data Augmentation. We study different data augmentation strategies for model editing.[8] We experiment with naïve fine-tuning, i.e., no augmentation, along with fine-tuning and SAUL with random augmentation (R), paraphrase augmentation (P), and both augmentations (P+R).

As shown in Table 4, fine-tuning with any data augmentation significantly improves the editing score compared to naive fine-tuning, but at the

[8]We follow the data augmentation strategies used in Gangadhar and Stratos (2024).

cost of generation quality. In particular, paraphrase augmentation causes a degradation of the model's generation quality, likely because it introduces unnatural sentence segments.[9] As shown in Figure 2, our method, SAUL w/ R, outperforms other methods in terms of generation fluency and consistency, and achieving strong editing scores across datasets.

6 Conclusion

In this work, we proposed SAUL, a novel finetuning method to address the challenges of preserving unrelated knowledge in LLMs and maintaining high generation quality during model editing. To achieve this, SAUL regularizes the generation process through sentence concatenation with augmented random facts. Our evaluation on three benchmark datasets demonstrated that SAUL outperforms state-of-the-art methods while maintaining generation quality and reducing computational overhead. Consequently, SAUL offers an efficient and practical solution for model editing in LLMs.

Limitations

Data Augmentation Strategies. Data augmentation is an active research area in natural language processing. In this work, we explore paraphrase and random augmentation to regularize the model's generation. Investigating additional data augmentation strategies could further improve performance and offer new insights into the model editing task, which we leave for future work.

Multilingual Model Editing Evaluation. Our evaluations are limited to monolingual datasets due to the absence of well-established multilingual datasets. To assess the effectiveness and generalizability of SAUL across diverse linguistic contexts, experiments with multilingual datasets are essential. This would help determine how well our method adapts to languages with various vocabulary sets and linguistic features.

Experiments with Different Numbers of Edits. In this work, we focus on the mass-editing setting following prior works (Meng et al., 2022b; Gangadhar and Stratos, 2024). Specifically, the CounterFact, ZsRE, and WikiRecent datasets used in this work provide 10,000, 10,000, and 1,266 requested edits, respectively. Investigating the performance and stability of SAUL under varying numbers of edits could provide valuable information about its scalability. This would be an interesting direction for future research.

[9] Please refer to Appendix A.1 for more details.

Ethical Considerations

One potential ethical issue of this work arises from the use of the CounterFact dataset which contains incorrect factual knowledge. While this dataset is valuable for testing and improving model editing methods, it inherently introduces the risk of propagating incorrect information if not carefully managed. Model editing based on such a dataset can inadvertently lead to the generation of incorrect information and hallucinated text.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive feedback. This work was partially supported by Deutsche Forschungsgemeinschaft (project SCHU 2246/14-1).

References

Hichem Ammar Khodja, Frederic Bechet, Quentin Brabant, Alexis Nasr, and Gwénolé Lecorvé. 2024. WikiFactDiff: A large, realistic, and temporally adaptable dataset for atomic factual knowledge update in causal language models. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Torino, Italia. ELRA and ICCL.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In Advances in Neural Information Processing Systems, volume 33, pages 1877–1901. Curran Associates, Inc.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. Transactions of the Association for Computational Linguistics, 12:283–298.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Govind Gangadhar and Karl Stratos. 2024. Model editing by pure fine-tuning. arXiv preprint arXiv:2402.11078.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2024. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. Advances in Neural Information Processing Systems, 36.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In International Conference on Learning Representations.

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

Nora Kassner, Philipp Dufter, and Hinrich Schütze. 2021. Multilingual LAMA: Investigating knowledge in multilingual pretrained language models. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 3250–3258, Online. Association for Computational Linguistics.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

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

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. In The Eleventh International Conference on Learning Representations.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. In International Conference on Learning Representations.

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

Ercong Nie, Bo Shao, Zifeng Ding, Mingyang Wang, Helmut Schmid, and Hinrich Schütze. 2024. Bmike-53: Investigating cross-lingual knowledge editing with in-context learning. arXiv preprint arXiv:2406.17764.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5418–5426, Online. Association for Computational Linguistics.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

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. 2024. 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, Bangkok, Thailand. Association for Computational Linguistics.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 10222–10240, Singapore. Association for Computational Linguistics.

Miaoran Zhang, Vagrant Gautam, Mingyang Wang, Jesujoba Alabi, Xiaoyu Shen, Dietrich Klakow, and Marius Mosbach. 2024. The impact of demonstrations on multilingual in-context learning: A multidimensional analysis. In Findings of the Association for Computational Linguistics ACL 2024, pages 7342–7371, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 4862–4876, Singapore. Association for Computational Linguistics.

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

A Appendix

A.1 Dataset Information

We evaluate SAUL and related methods on three datasets: CounterFact, ZsRE, and WikiRecent.[10] CounterFact (Meng et al., 2022b) is a dataset that includes artificially created counterfacts to test the ability of model editing methods to add counterfactual information to the language model. ZsRE (Levy et al., 2017) is a question-answering dataset consisting of 10,000 real-world facts, used to test model editing methods for adding correct information. WikiRecent (Cohen et al., 2024) collects factual knowledge that has been inserted into WikiData after July 2022.

Specifically, the CounterFact, ZsRE, and WikiRecent datasets provide 10,000, 10,000, and 1,266 requested edits, respectively. For each requested edit, we augment 20 unrelated true facts provided by Gangadhar and Stratos (2024) for sentence concatenation. For the data augmentation ablation study, we add paraphrase samples for augmentation following Gangadhar and Stratos (2024). They augment the paraphrase data by generating free texts using the GPT-J model and prepend these texts to the original factual sentence for model editing. The generated sentence segments are listed in Table 5. As discussed in Section 5, paraphrase augmentation causes a degradation in the model's generation quality, likely because it introduces unnatural sentences such as "Q: How can I use a. The mother tongue of Danielle Darrieux is English".

Paraphrase prefix
"Q: . "
"Q: . "
"The present invention relates."
"The role of the."
"\n \n-."
"Q: Why is my code not."
"Q: What is the correct way."
"The present invention relates in general to the manufacture."
"The role of the family in the development of."
"\n \n-\n \n1\n."
"A new report from the Center for Immigration Studies."
"Q: How can I use a."
"Q: How to use multiple variables."
"\n \n=\n \n1\n."
"Q: What is the difference in."

Table 5: Examples of the prefix text used for paraphrase augmentation.

[10]We select the datasets following previous works (Mitchell et al., 2021; Meng et al., 2022a,b; Gangadhar and Stratos, 2024), and leave the extension to other model editing datasets, such as Zhong et al. (2023); Ammar Khodja et al. (2024); Nie et al. (2024), for future work.

A.2 Implementation Details

We use the AdamW optimizer (Loshchilov and Hutter, 2017) for all experiments.Table 6 provides detailed hyperparameter choices for SAUL across datasets. The training was performed on Nvidia A100 GPUs.[11]

	CounterFact	ZsRE	WikiRecent
Epochs		40	
Early stop patience		5	
Batch size		32	
No. augmented facts	20	20	10
Learning rate	5e-5	2e-5	1e-4

Table 6: Hyperparameters used on three model editing datasets used in this work.

A.3 Additional Experimental Results

As introduced in Section 4, model editing performance is evaluated using *efficacy*, *generality*, and *locality*. In Section 5, we report the harmonic mean of these three metrics in the main paper for brevity. Here in Table 7 to 16, we provide the complete evaluation results, including all these model editing metrics and the generation metrics *fluency* and *consistency*. Here, we also include the experimental results of the in-context knowledge editing (IKE) method (Zheng et al., 2023), which allows the model to acquire new knowledge directly from the input context (Brown et al., 2020; Zhang et al., 2024). [12] It is important to note that our work focuses on gradient-based model editing in the mass-editing setting, where multiple facts are edited simultaneously. In contrast, knowledge editing with in-context learning is limited to the single-edit case, making it an unsuitable baseline for our approach. Nonetheless, we include the IKE results to offer a more comprehensive comparison and to highlight SAUL's relative strengths.

[11]All experiments ran on a carbon-neutral GPU cluster.

[12]We experiments use the IKE implementation in EasyEdit (Wang et al., 2024).

Editor	CounterFact					
	Score	*Efficacy*	*Generality*	*Locality*	Fluency	Consistency
Original GPT-J	22.4	15.2	17.7	83.5	622.4	29.4
MEND	23.1	15.7	18.5	83.0	618.4	31.1
ROME	50.3	50.2	50.4	50.2	589.6	3.3
MEMIT	85.8	98.9	88.6	73.7	**619.9**	**40.1**
IKE	74.3	100.0	95.1	50.3	620.9	29.2
FT + R + P	86.5	98.8	93.6	72.0	352.0	5.2
FT + R + P*	86.6	98.1	95.1	71.8	208.7	4.7
SAUL	**87.7**	99.6	92.8	74.8	600.7	31.0

Table 7: Complete evaluation results on CounterFact of SAUL and related methods on three benchmark datasets.

Editor	ZsRE				
	Score	*Efficacy*	*Generality*	*Locality*	Fluency
Original GPT-J	26.4	26.4	25.8	27.0	599.0
MEND	20.0	19.4	18.6	22.4	-
ROME	2.6	21.0	19.6	0.9	-
MEMIT	50.7	96.7	89.7	26.6	-
IKE	**65.4**	100.0	98.7	38.9	584.6
FT + R + P	62.0	99.9.	97.0	35.6	-
FT + R + P*	64.2	97.0	87.2	40.1	591.5
SAUL	63.6	99.9	93.4	37.8	**620.7**

Table 8: Complete evaluation results on ZsRE of SAUL and related methods on three benchmark datasets.

Editor	WikiRecent				
	Score	*Efficacy*	*Generality*	*Locality*	Fluency
Original GPT-J	37.4	34.4	34.5	45.3	600.8
MEND	-	-	-	-	-
ROME	35.0	39.8	25.5	46.9	-
MEMIT	67.3	99.2	80.2	45.3	-
IKE	**77.8**	100.0	85.4	54.3	**574.5**
FT + R + P	68.5	99.6	84.6	45.8	-
FT + R + P*	70.1	99.6	93.4	45.4	501.3
SAUL	69.7	99.5	89.1	46.0	560.6

Table 9: Complete evaluation results on WikiRecent of SAUL and related methods on three benchmark datasets.

Editor	CounterFact					
	Score	*Efficacy*	*Generality*	*Locality*	Fluency	Consistency
Original GPT-J	22.4	15.2	17.7	83.5	622.4	29.4
FT 21st	57.0	84.3	52.0	46.5	**584.4**	**14.9**
FT 3-8th	60.8	99.9	82.5	36.8	553.8	8.7
FT all	**62.4**	99.9	91.2	36.9	452.1	4.3
FT LoRA	55.4	100.0	71.6	33.1	494.4	5.7
SAUL 3-8th	**89.8**	99.5	92.4	79.7	595.4	30.1
SAUL all	87.7	99.6	92.8	74.6	**600.7**	**31.0**

Table 10: Complete evaluation results on CounterFact for the ablation study with various fine-tuning paradigms.

Table 11: Complete evaluation results on ZsRE for the ablation study with various fine-tuning paradigms.

Editor	ZsRE				
	Score	*Efficacy*	*Generality*	*Locality*	Fluency
Original GPT-J	26.4	26.4	25.8	27.0	599.0
FT 21st	37.9	45.7	43.4	29.2	**566.4**
FT 3-8th	56.7	98.9	96.5	30.9	549.5
FT all	**58.8**	99.5	96.3	32.7	559.9
FT LoRA	57.8	96.5	92.4	32.6	543.9
SAUL 3-8th	**63.6**	99.7	85.1	39.4	615.0
SAUL all	**63.6**	99.9	93.4	37.8	**620.7**

Table 12: Complete evaluation results on ZsRE for the ablation study with various fine-tuning paradigms.

Editor	WikiRecent				
	Score	*Efficacy*	*Generality*	*Locality*	Fluency
Original GPT-J	37.4	34.4	34.5	45.3	600.8
FT 21st	45.7	48.8	43.7	45.0	**595.8**
FT 3-8th	**69.2**	99.6	87.8	45.5	574.3
FT all	67.2	99.6	79.8	45.3	570.0
FT LoRA	67.5	99.4	81.4	45.3	546.8
SAUL 3-8th 3-8th	69.4	99.5	85.5	46.5	**587.9**
SAUL 3-8th all	**69.7**	99.5	89.1	46.0	560.6

Table 13: Complete evaluation results on WikiRecent for the ablation study with various fine-tuning paradigms.

Editor	CounterFact					
	Score	*Efficacy*	*Generality*	*Locality*	Fluency	Consistency
Original GPT-J	22.4	15.2	17.7	83.5	622.4	29.4
FT	62.4	99.9	91.2	36.9	452.1	4.3
FT + R	85.3	98.7	87.6	73.5	379.0	3.5
FT + P	70.7	99.9	99.2	44.7	190.9	5.6
FT + P + R	86.6	98.1	95.1	71.8	208.7	4.7
SAUL w/ R	**87.7**	99.6	92.8	74.6	**600.7**	**31.0**
SAUL w/ P	68.7	100.0	97.4	42.7	366.8	8.6
SAUL w/ P + R	87.5	99.8	92.1	74.5	447.6	18.0

Table 14: Complete evaluation results on CounterFact for the ablation study with various data augmentation strategies.

Editor	ZsRE				
	Score	*Efficacy*	*Generality*	*Locality*	Fluency
Original GPT-J	26.4	26.4	25.8	27.0	599.0
FT	58.8	99.5	96.3	32.7	559.9
FT + R	58.6	99.6	98.5	32.2	564.2
FT + P	63.7	99.8	94.2	37.8	607.2
FT + P + R	**64.2**	97.0	87.2	40.1	591.5
SAUL w/ R	63.6	99.9	93.4	37.8	**620.7**
SAUL w/ P	54.4	99.9	96.0	28.8	466.9
SAUL w/ P + R	63.5	99.9	94.9	37.4	490.3

Table 15: Complete evaluation results on ZsRE for the ablation study with various data augmentation strategies.

Editor	WikiRecent				
	Score	*Efficacy*	*Generality*	*Locality*	Fluency
Original GPT-J	37.4	34.4	34.5	45.3	600.8
FT	67.2	99.6	79.8	45.3	**570.0**
FT + R	69.9	99.6	92.2	45.4	454.6
FT + P	69.0	99.5	85.4	46.1	541.5
FT + P + R	70.1	99.6	93.4	45.4	501.3
SAUL w/ R	69.7	99.5	89.1	46.0	560.6
SAUL w/ P	69.5	99.5	87.7	46.1	406.4
SAUL w/ P + R	**70.5**	99.5	86.7	47.7	437.8

Table 16: Complete evaluation results on WikiRecent for the ablation study with various data augmentation strategies.