Persuasion Tokens for Editing Factual Knowledge in LLMs

Paul Youssef **Christin Seifert** **Jörg Schlötterer**
Marburg University
{paul.youssef, christin.seifert, joerg.schloetterer}@uni-marburg.de

Abstract

In-context knowledge editing (IKE) is a promising technique for updating Large Language Models (LLMs) with new information. However, IKE relies on lengthy, fact-specific demonstrations which are costly to create and consume significant context window space. In this paper, we introduce persuasion tokens (`P-Tokens`) – special tokens trained to replicate the effect of IKE demonstrations, enabling efficient knowledge editing without requiring fact-specific demonstrations. We evaluate `P-Tokens` across two editing datasets and three LLMs, demonstrating performance comparable to, and often exceeding, IKE. We further find that editing performance is robust to distractors with small negative effects to neighboring facts, and that increasing the number of `P-Tokens` improves performance. Our work addresses key limitations of IKE and provides a more practical and scalable alternative for editing LLMs.[1]

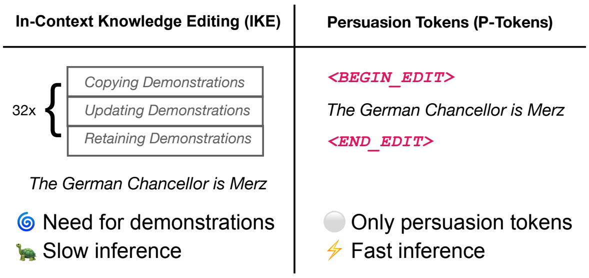

Figure 1: In-context knowledge editing (IKE) relies on complex demonstrations and leads to slower inference. Persuasion tokens (`P-Tokens`) eliminate the need for long demonstrations and lead to faster inference.

1 Introduction

Large Language Models (LLMs) encode facts about the world in their parameters (Petroni et al., 2019; Youssef et al., 2023, 2024a). Knowledge editing methods (KEs) have been introduced to address the problem of outdated factual knowledge in LLMs (Wang et al., 2024b; Mazzia et al., 2024). KEs differ in how they update knowledge in LLMs. Parameter-modifying KEs (Meng et al., 2023; Tan et al., 2023) directly change the model's parameters, whereas parameter-preserving KEs (Wang et al., 2024a; Guo et al., 2025) introduce additional memory modules to update knowledge without affecting the original parameters.

In-context knowledge editing (IKE) (Zheng et al., 2023) makes use of the strong in-context learning abilities of LLMs and updates knowledge in LLMs by adding the updated knowledge as input to the model. More specifically, IKE constructs a prompt that contains the new fact with a set of 32 demonstrations to teach the model to use the new fact in semantically similar contexts, and to not affect other irrelevant facts. IKE has been shown to have strong performance. The diverse demonstrations, which are the reason for the strong performance of IKE, bring also key limitations. These demonstrations are fact-specific and have to be constructed from scratch for each edit. These demonstrations also make the prompt longer, which consumes much of the context window and makes inference slower.

We introduce persuasion tokens (`P-Tokens`), which are special tokens that are tuned to have the same effect as the IKE demonstrations and can edit facts in-context. `P-Tokens` eliminate the need for fact-specific demonstrations and replace the long IKE edits, leading to faster inference (see Figure 1). We train and evaluate `P-Tokens` on two standard knowledge editing datasets and three LLMs. Our experiments show strong performance of `P-Tokens` compared to IKE across all settings. Performance improves, with increasing the number of `P-Tokens`. We further study the effect of distractors (additional edits) on the editing performance. We find that the

[1] https://github.com/paulyoussef/p-tokens

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 2: Short Papers, pages 475–486

performance on editing and paraphrased prompts remains robust to distractors, but neighboring facts are negatively affected. We further compare the number of tokens and inference time for IKE and P-Tokens. Compared to IKE, P-Tokens require less tokens and have less inference time, highlighting the efficiency of P-Tokens. Our work provides an efficient, effective and a more practical alternative to IKE, paving the way for wider adoption.

2 Background and Problem Statement

In-context Knowledge Editing. Facts are often represented as triplets of $(subject, relation, object)$, or (s, r, o) for short. Querying an LLM with a prompt $p(s, r)$, where p expresses the relation r with the subject s (e.g., "In which city is the Eiffel Tower?"), should lead to generating the object o (e.g., "Paris"), given that the fact (s, r, o) is encoded in the LLM. A knowledge editing operation $E(s, r, o, o', p)$ is successful if it changes the output of the LLM such that the retrieved object is o' instead of o. IKE edits are conducted by prepending an editing prompt $p_{IKE}(s, r, o')$ to the query prompt $p(s, r)$, i.e., $p_{IKE}(s, r, o') \oplus p(s, r)$, where $\oplus$ is the string concatenation operation. The editing prompt $p_{IKE}(s, r, o')$ causes the LLM's output to change from o to o'.

Problem Statement. Given a model $\mathcal{M}$ that outputs the object o, when provided with a prompt $p(s, r)$, i.e., $o = argmax_{o*}\ \mathbb{P}_{\mathcal{M}(p(s,r))}[o^*]$, where $\mathbb{P}_{\mathcal{M}(p(s,r))}$ is the model's output probability distribution over the vocabulary $\mathcal{V}$ given the prompt $p(s, r)$ and $o^* \in \mathcal{V}$. $\mathcal{M}$'s output is changed to o' with IKE (Zheng et al., 2023) by prepending an IKE editing prompt $p_{IKE}(s, r, o')$ (see Figure 3 in the appendix for an example) to the original prompt, i.e., $o' = argmax_{o^*}\ \mathbb{P}_{\mathcal{M}(p_{IKE}(s,r,o') \oplus p(s,r))}[o^*]$. Our goal is to replace the editing prompt $p_{IKE}(s, r, o')$ with a significantly shorter prompt $p_{PT}(s, r, o')$ such that $o' = argmax_{o^*}\ \mathbb{P}_{\mathcal{M}(p_{PT}(s,r,o') \oplus p(s,r))}[o^*]$. Accordingly, the shorter prompt $p_{PT}(s, r, o')$ should have the same effect as the IKE editing prompt $p_{IKE}(s, r, o')$, and lead to generating o'.

3 Method

To replace the IKE prompt with a short and efficient editing prompt, we enclose the edit $p(s, r, o')$ with *special editing tokens:* BEGIN_EDIT and END_EDIT. We refer to these tokens as *persuasion tokens* or P-Tokens for short. We further optimize the embedding vectors of these tokens to minimize the Kullback-Leibler (KL) divergence loss between two output distributions:

$$\mathcal{L} = KL[P_{PT} || P_{IKE}] \tag{1}$$

where $P_{PT} = \mathbb{P}_{\mathcal{M}(p_{PT}(s,r,o') \oplus p(s,r))}$ is the output distribution of the model when using the above described editing prompt with P-Tokens, and $P_{IKE} = \mathbb{P}_{\mathcal{M}(p_{IKE}(s,r,o') \oplus p(s,r))}$ is the output distribution when using the long IKE editing prompt. To improve performance, we further minimize the KL-divergence between the output distributions that correspond to the following pairs of inputs:

Paraphrases: $p_{PT}(s, r, o') \oplus p'(s, r)$ and $p_{IKE}(s, r, o') \oplus p'(s, r)$, where $p'(s, r)$ refers to a paraphrased version of the prompt $p(s, r)$.

Neighbors: $p_{PT}(s, r, o') \oplus p(\bar{s}, r)$ and $p(\bar{s}, r)$. Here, we use $p(\bar{s}, r)$ to refer to neighboring prompts, that query similar facts to the one targeted by the edit (same relation, but different subject). Our goal is to avoid affecting irrelevant facts that should not be changed by the edit.

Distractors: We add a distractor that consists of several edits (with P-Tokens) between the edit $p_{PT}(s, r, o')$, and the querying prompt $p(s, r)$, to enhance robustness.

Other: We add P-Tokens to other prompts that do not include any edits, such that P-Tokens have no effect when there are no edits.

We provide an overview of all of the pairs of inputs in Tables 5 and 6 in Appendix A. Each training batch includes all types of input pairs.

4 Experimental Setup

Datasets. We use two datasets in our experiments. The first is CounterFact (Meng et al., 2022), which contains counterfactual statements (e.g., "The space needle is located in Rome."). We use the same subset of 2,000 examples as Zheng et al. (2023). We use 800 of these examples for training, 200 for validation and the remaining 1,000 for testing. The second dataset is zsRE (Levy et al., 2017; Mitchell et al., 2022a), which contains question-answer pairs. From the training set, we randomly sample 800 instances for training and 200 for validation. For testing, we use the whole test set of 19,086 examples. There are no IKE edits for this dataset. As a replacement for IKE edits, we use a baseline containing the editing prompt or the paraphrased prompt. An overview of the examples used for zsRE is shown in Table 6 in Appendix A.

Evaluation. We follow previous work (Meng et al., 2022, 2023) in using dataset-specific metrics. For CounterFact, the evaluation is based on the probability of the original object o and the edited object o'. An edit is considered successful if the probability of o' is higher than the probability of o. The metric **ES** refers to the editing success rate when the editing prompts and the prompts to retrieve the edited fact are the same, whereas **PS** refers to the success rate when the prompts to retrieve the edited fact are paraphrased. A corresponding metric for neighboring facts is **NS**, where the goal is to not affect these prompt by retaining a higher probability for o than o'. The overall performance is summarized by **S**, which is the harmonic mean of **ES**, **PS** and **NS**. For zsRE, the metrics are based on the accuracy of the model in generating o' using the editing prompts (**Efficacy**), the paraphrased prompts (**Paraphrase**), and the accuracy of the model on unrelated facts (**Specificity**). We provide more formal definitions in Appendix A.

Baselines. We compare against IKE (Zheng et al., 2023) to asses if `P-Tokens` are as effective. IKE demonstrations are only available for CounterFact. Therefore, we cannot report IKE performance on zsRE. We instead compare against a baseline that contains the edit $p(s, r, o')$ without `P-Tokens`, and also report results on the original model without any edits.

Number of `P-Tokens`. To analyze how the number of tokens affects performance, we experiment with m persuasions tokens, $m \in \{1, 3, 5, 7, 10\}$. This means, we use m `BEGIN_EDIT` tokens and m `END_EDIT` tokens. Each of the $2m$ tokens has its own separate representation.

Models. We use GPT-J-6B (Wang and Komatsuzaki, 2021), Qwen2.5-7B, Qwen2.5-14B (Team, 2024) and Llama3-8B (Grattafiori et al., 2024).

5 Results and Discussion

The results on CounterFact are shown in the left part of Table 1. We report the maximum performance across different numbers of `P-Tokens`. More detailed results can be found in Appendix B. `P-Tokens` outperform IKE across all models. For example, the **S** value on Llama3 with IKE is 93.97, whereas with `P-Tokens` it is 95.82. On Qwen-7B, the gap is even larger ($\approx$ 5 p.p.). Generally, the performance of `P-Tokens` is on par or slightly better than the performance of IKE with respect to **ES** and **PS**. This shows that `P-Tokens` are as effective as IKE in editing knowledge. We notice a larger gap with respect to **NS** ($\approx$ 4 p.p. on Llama3, Qwen-14B and GPT-J, and $\approx$ 10 p.p. on Qwen-7B), which indicates that `P-Tokens` have minimum effects on neighboring facts. We attribute this to our optimization criteria (cf. Table 5 in Appendix A), where we optimize the outputs with `P-Tokens` on unrelated facts to be the same as the outputs of the original model, which also explains the small gap to the original model (1.58 p.p. at most).

The right part of Table 1 shows the results for zsRE. We observe that `P-Tokens` outperform the baseline in most cases. The gaps in performance differ across metrics. With **Efficacy**, the differences vary between $\approx$ 1 p.p. (Llama3) and $\approx$ 4 p.p. (Qwen and GPT-J). With **Paraphrase** the differences are larger and vary between $\approx$ 4 p.p. (Llama3) and $\approx$ 10 p.p. (Qwen and GPT-J). The high **Paraphrase** values are due to our optimization criteria (cf. Table 6 in Appendix A), where we target approximating a model's output, whose input includes the edit with a paraphrased prompt. With **Specificity** the values differ slightly from the values of the original models, indicating minor effects to neighboring facts.

Number of `P-Tokens`. We show how the performance varies across different numbers of `P-Tokens` for CounterFact and zsRE in Figure 2. On CounterFact, we notice that Qwen-7B benefits the most from increasing the number of tokens. Qwen-7B's **PS** and **NS** increase by $\approx$ 5 p.p. when using 3 instead of 1 tokens. On zsRE, Llama3's **Efficacy** and **Paraphrase** increase the most by $\approx$ 7 and 16 p.p. respectively, when increasing the number of tokens from 1 to 3. Generally, the results show that editing performance improves when increasing the number of tokens.

Effect of distractors. We evaluate the performance of `P-Tokens` with distractors of varying length on a subset of 1,000 facts from CounterFact. The results in Table 2 show that **ES** and **PS** scores remain high, indicating successful editing on both the editing and paraphrased prompts. The overall performance (**S** score) drops by $\approx$ 5 and

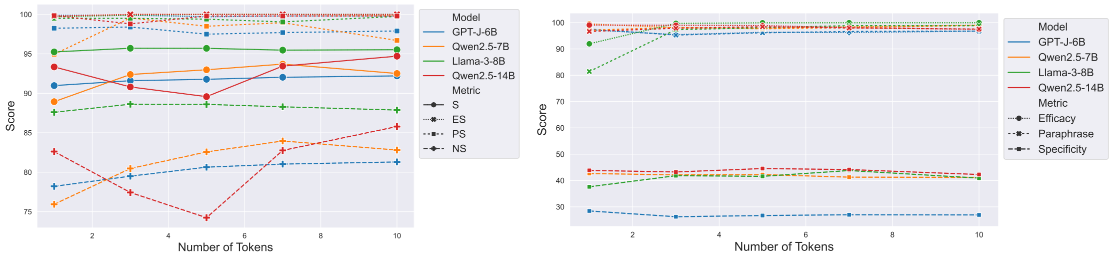

Figure 2: Performance across different numbers of P-Tokens (Left: CounterFact, Right: zsRE). On CounterFact, Qwen2.5-7B benefits from increasing the number of tokens, especially on **PS** and **NS**. On zsRE, Llama3's **Efficacy** and **Paraphrase** increase when the number of P-Tokens increases.

		CounterFact				**zsRE**		
Model	**KE**	**S**	**ES**	**PS**	**NS**	**Efficacy**	**Paraphrase**	**Specificity**
GPT-J	original	24.49	17.10	19.25	**82.70**	27.23	26.43	27.22
	baseline	74.02	95.90	87.90	53.41	93.36	87.96	25.39
	IKE	89.96	**100**	97.30	76.51	—	—	—
	`P-Tokens`	**92.39**	99.90	**98.40**	81.30	**97.29**	**97.54**	**28.44**
Qwen-7B	original	21.45	13.80	17.95	**85.54**	39.50	37.91	39.51
	baseline	62.07	99.80	90.40	36.70	96.41	89.16	**44.18**
	IKE	89.08	**100**	98.75	73.79	—	—	—
	`P-Tokens`	**93.88**	**100**	**99.55**	83.96	**99.51**	**98.94**	42.66
Llama3	original	12.14	7.50	9.75	**89.36**	38.46	37.19	39.15
	baseline	72.42	**100**	89.35	49.43	98.73	95.14	43.49
	IKE	93.97	**100**	99.25	84.40	—	—	—
	`P-Tokens`	**95.82**	**100**	**99.75**	88.62	**99.96**	**98.85**	**43.80**
Qwen-14B	original	17.27	10.60	14.70	**87.82**	42.34	40.74	42.16
	baseline	63.86	99.90	90.65	38.56	94.01	87.51	**45.04**
	IKE	92.84	**100**	99.30	81.68	—	—	—
	`P-Tokens`	**94.72**	**100**	**99.85**	85.79	**99.07**	**98.44**	44.55

Table 1: Performance on CounterFact and zsRE. `P-Tokens` outperform IKE and the baseline across all models.

7 p.p. on Qwen and Llama3 respectively, while it remains relatively stable on GPT-J. The **ES** and **PS** scores remain high indicating successful editing on both the editing and paraphrased prompts. However, the **NS** scores (and accordingly the overall scores **S**) drop significantly on Qwen-7B (10 p.p.), Qwen-14B (14 p.p.) and Llama3 (15 p.p.). This drop suggests that distractors mostly affect neighboring facts. A potential remedy might be increasing the distractor length used in training (we use distractors consisting of 5 or 10 edits).

Ablation. We retrain `P-Tokens` for Llama3 without distractors to verify their effect on the performance. The results in Table 3 show that the performance of `P-Tokens` drops when training without distractors. For example, the **S** score drops from 89.55 (cf. Table 2) to 82.94 when having a distractor of 100 edits, which demonstrates the effectiveness of having distractors during training for a more robust performance.

Efficiency. Part of the motivation for `P-Tokens` is to make inference more efficient. We compare the number of tokens and the inference time for IKE and `P-Tokens` using Qwen-7B with 10×2 `P-Tokens`, a sample of 1,000 edits and a batch size of 1 for both methods. The results in Table 4 show that IKE prompts are more than 16 times as long as `P-Tokens` prompts. Inference with `P-Tokens` is over five times faster than with IKE. The shorter prompts of `P-Tokens` also enable larger batch sizes, increasing the inference time advantage even more.

Amortization point. Despite the advantages at inference time, `P-Tokens` add costs for the initial training. We calculate at which point the training

Model	|Dist.|	S	ES	PS	NS
GPT-J	0	92.79	99.80	98.05	82.57
	10	92.86	99.90	98.55	82.31
	50	92.18	99.80	98.90	80.55
	100	92.37	99.90	98.85	80.95
Qwen-7B	0	94.10	100	99.45	84.55
	10	92.76	100	99.65	81.27
	50	90.85	100	99.55	77.06
	100	89.86	100	99.40	75.04
Llama3	0	96.36	100	99.55	90.18
	10	95.59	100	99.80	88.00
	50	93.59	100	99.30	83.44
	100	89.55	100	98.80	74.75
Qwen-14B	0	94.88	100.0	99.90	86.15
	10	93.72	100.0	99.90	83.33
	50	91.81	100.0	99.80	79.01
	100	88.87	100.0	99.90	72.75

Table 2: Performance of P-Tokens on a subset of CounterFact when adding distractors of varying length. |**Dist.**| refers to the distractor length. **ES** and **PS** remain high, while **NS** drops, indicating that neighbouring facts are affected more as the distractor becomes longer.

Model	|Dist.|	S	ES	PS	NS
Llama3	0	96.33	100.0	99.35	90.28
	10	94.75	99.9	95.80	89.18
	50	87.93	98.4	84.20	82.78
	100	82.94	91.9	80.25	77.95

Table 3: Performance of P-Tokens, trained *without* distractors, on a subset of CounterFact when adding distractors of varying length. The performance of P-Tokens drops compared to our proposed setup (cf. Table 2).

Method	#Tokens	Inference Time per Edit
IKE	959.19	0.17
P-Tokens	58.01	0.03

Table 4: Comparison of IKE and P-Tokens with respect to the number of tokens and inference time in seconds, using Qwen2.5-7B with 10×2 P-Tokens and a sample of 1000 edits. Inference with P-Tokens is over five times faster than with IKE.

costs amortize with Qwen-7B. Training Qwen-7B with 10×2 P-Tokens takes roughly 15 hours and 28 minutes. This means that P-Tokens amortize after $(15 \times 60 \times 60 + 28 \times 60)/(0.17 - 0.03) = 398$k inferences. If the expected number of inferences is lower, training P-Tokens is not economical, but still has other advantages like eliminating the need for edit-specific demonstrations and requiring shorter prompts.

6 Related Work

Knowledge Editing methods (KEs) (Wang et al., 2024b; Mazzia et al., 2024) can be classified as either parameter-modifying or parameter-preserving. Parameter-modifying KEs directly change the model's parameters to update facts, and can further be categorized as locate-and-edit KEs (Meng et al., 2022, 2023), that first locate the parameters responsible for the facts and then adapt these parameters, or meta-learning KEs (Mitchell et al., 2022a; Tan et al., 2023) that train hypernetworks to predict the necessary shift in parameters to update facts. Parameter-preserving KEs add special memory modules that update the targeted facts (Mitchell et al., 2022b; Hartvigsen et al., 2023; Wang et al., 2024a; Guo et al., 2025), or make use of the model's in-context abilities (Cohen et al., 2024; Youssef et al., 2024b) to update some facts using some demonstrations (Zheng et al., 2023). Our work eliminates the need for fact-specific demonstrations and makes editing easier and more efficient using P-Tokens.

7 Conclusion

In this work, we introduced persuasion tokens as an alternative to in-context knowledge edits that require many demonstrations and overload the model's context. We showed that persuasion tokens outperform in-context edits on two datasets and three LLMs. We further showed that increasing the number of persuasion tokens has a positive effect on the performance, and that distractors negatively affect the performance on neighboring facts, while the performance on editing and paraphrased prompts remain robust to distractors. P-Tokens are an efficient and effective alternative to IKE, and their integration in LLMs enables easy editing.

Limitations

Persuasion tokens can be used as an alternative to IKE demonstrations and lead to faster inference. However, these tokens need to be trained initially, which incurs additional computational costs. Knowledge Editing is shown to have potential malicious use cases (Chen et al., 2024; Li et al., 2025; Cheng et al., 2025; Youssef et al., 2025a,c,d).

P-Tokens could make prompt injection attacks easier due to the prompts being shorter. However, some work already addresses this risk (Youssef et al., 2025b; Chen et al., 2025).

Acknowledgments

We gratefully acknowledge support from the hessian.AI Service Center (funded by the Federal Ministry of Research, Technology and Space, BMFTR, grant no. 16IS22091) and the hessian.AI Innovation Lab (funded by the Hessian Ministry for Digital Strategy and Innovation, grant no. S-DIW04/0013/003)

References

Canyu Chen, Baixiang Huang, Zekun Li, Zhaorun Chen, Shiyang Lai, Xiongxiao Xu, Jia-Chen Gu, Jindong Gu, Huaxiu Yao, Chaowei Xiao, Xifeng Yan, William Yang Wang, Philip Torr, Dawn Song, and Kai Shu. 2024. Can Editing LLMs Inject Harm? In *ICML 2024 Next Generation of AI Safety Workshop*.

Sizhe Chen, Yizhu Wang, Nicholas Carlini, Chawin Sitawarin, and David Wagner. 2025. Defending Against Prompt Injection with a Few DefensiveTokens. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*.

Yinjie Cheng, Paul Youssef, Christin Seifert, Jörg Schlötterer, and Zhixue Zhao. 2025. Can Fine-Tuning Erase Your Edits? On the Fragile Coexistence of Knowledge Editing and Adaptation. *Preprint*, arXiv:2511.05852.

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.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Dongliang Guo, Mengxuan Hu, Zihan Guan, Thomas Hartvigsen, and Sheng Li. 2025. BalancEdit: Dynamically Balancing the Generality-Locality Trade-off in Multi-modal Model Editing. In *Forty-second International Conference on Machine Learning*.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with GRACE: Lifelong Model Editing with Discrete Key-Value Adaptors. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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.

Xiaopeng Li, Shasha Li, Shangwen Wang, Shezheng Song, Bin Ji, Huijun Liu, Jun Ma, and Jie Yu. 2025. Identifying Knowledge Editing Types in Large Language Models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 1553–1564, New York, NY, USA. Association for Computing Machinery.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Vittorio Mazzia, Alessandro Pedrani, Andrea Caciolai, Kay Rottmann, and Davide Bernardi. 2024. A Survey on Knowledge Editing of Neural Networks. *Preprint*, arXiv:2310.19704.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. 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. 2023. 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. 2022a. Fast Model Editing at Scale. In *International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022b. Memory-Based Model Editing at Scale. In *International Conference on Machine Learning*.

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.

Chenmien Tan, Ge Zhang, and Jie Fu. 2023. Massive Editng for Large Language Models via Meta Learning. *arXiv preprint arXiv:2311.04661*.

Qwen Team. 2024. Qwen2.5: A Party of Foundation Models.

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, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024a. WISE: Rethinking the Knowledge Memory for Lifelong Model Editing of Large Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024b. Knowledge Editing for Large Language Models: A Survey. *ACM Comput. Surv.*, 57(3).

Paul Youssef, Osman Koraş, Meijie Li, Jörg Schlötterer, and Christin Seifert. 2023. Give Me the Facts! A Survey on Factual Knowledge Probing in Pre-trained Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15588–15605, Singapore. Association for Computational Linguistics.

Paul Youssef, Jörg Schlötterer, and Christin Seifert. 2024a. Enhancing Fact Retrieval in PLMs through Truthfulness. *arXiv preprint arXiv:2410.13562*.

Paul Youssef, Jörg Schlötterer, and Christin Seifert. 2024b. The Queen of England is not England's Queen: On the Lack of Factual Coherency in PLMs. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2342–2354, St. Julian's, Malta. Association for Computational Linguistics.

Paul Youssef, Zhixue Zhao, Daniel Braun, Jörg Schlötterer, and Christin Seifert. 2025a. Position: Editing Large Language Models Poses Serious Safety Risks. In *Forty-second International Conference on Machine Learning Position Paper Track*.

Paul Youssef, Zhixue Zhao, Jörg Schlötterer, and Christin Seifert. 2025b. How to Make LLMs Forget: On Reversing In-Context Knowledge Edits. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12656–12669, Albuquerque, New Mexico. Association for Computational Linguistics.

Paul Youssef, Zhixue Zhao, Christin Seifert, and Jörg Schlötterer. 2025c. Has this Fact been Edited? Detecting Knowledge Edits in Language Models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9768–9784, Albuquerque, New Mexico. Association for Computational Linguistics.

Paul Youssef, Zhixue Zhao, Christin Seifert, and Jörg Schlötterer. 2025d. Tracing and Reversing Rank-One Model Edits. *Preprint*, arXiv:2505.20819.

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.

A Evaluation and Implementation Details

Evaluation. We follow previous work (Meng et al., 2022, 2023) in using dataset-specific metrics. For CounterFact, we use the following metrics:

- Efficacy Score (**ES**): the percentage of facts where the probability of the edited object is higher than the probability of the original object $ES = \frac{1}{n}\sum_i^n \mathbb{1}[\mathbb{P}_{\mathcal{M}(p_{PT}(s,r,o')\oplus p(s,r))}(o') > \mathbb{P}_{\mathcal{M}(p_{PT}(s,r,o')\oplus p(s,r))}(o)]$ where $\mathbb{1}[\cdot]$ is the indicator function.

- Paraphrase Score (**PS**): Similar to ES, but with paraphrased prompts, i.e., $PS = \frac{1}{n}\sum_i^n \mathbb{1}[\mathbb{P}_{\mathcal{M}(p_{PT}(s,r,o')\oplus p'(s,r))}(o') > \mathbb{P}_{\mathcal{M}(p_{PT}(s,r,o')\oplus p'(s,r))}(o)]$.

- Neighborhood Score (**NS**): Proportion of neighboring (irrelevant) facts, for which the probability of the original object is still higher than that of the edited object, $NS = \frac{1}{n}\sum_i^n \mathbb{1}[\mathbb{P}_{\mathcal{M}(p_{PT}(s,r,o')\oplus p(\bar{s},r))}(o') < \mathbb{P}_{\mathcal{M}(p_{PT}(s,r,o')\oplus p(\bar{s},r))}(o)]$.

For zsRE, we measure the accuracy of the model in generating the edited object using editing (**Efficacy**) and paraphrased prompts (**Paraphrase**), $\frac{1}{n}\sum_i^n \mathbb{1}[o' = argmax_{o^*}\mathbb{P}_{\mathcal{M}(p_{PT}(s,r,o')\oplus p/p'(s,r))}(o*)]$, where $p/p'(s,r)$ refers to using either $p(s,r)$ or $p'(s,r)$. We measure the accuracy on unrelated facts (**Specifcity**) $\frac{1}{n}\sum_i^n \mathbb{1}[\bar{o} = argmax_{o^*}\mathbb{P}_{\mathcal{M}(p_{PT}(s,r,o')\oplus p(\bar{s},\bar{r}))}(o*)]$.

Optimization pairs. We show the optimization prompt pairs for CounterFact and zsRE in Table 5 and Table 6 respectively.

Implementation details. We optimize the embeddings of the `P-Tokens` using AdamW (Loshchilov and Hutter, 2019) optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$ with weight decay=0.01. We train for a maximum of 50 epochs using early stopping with with a patience of 3 epochs on the validation set. We use a distractor consisting of 5 or 10 facts for each edit

Datasets. CounterFact (Meng et al., 2022) is published under the MIT License. We could not find the license for zsRE (Levy et al., 2017).

Name	Persuasion Token Prompts	Target Prompts
Editing prompts	$p_{PT}(s,r,o') \oplus p(s,r)$	$p_{IKE}(s,r,o') \oplus p(s,r)$
Paraphrases	$p_{PT}(s,r,o') \oplus p'(s,r)$	$p_{IKE}(s,r,o') \oplus p'(s,r)$
Neighbors	$p_{PT}(s,r,o') \oplus p(\bar{s},r)$	$p(\bar{s},r)$
Distractor	$p_{PT}(s,r,o') \oplus p_{PT}(\bar{s},\bar{r},\bar{o}') \oplus p(s,r)$	$p_{IKE}(s,r,o') \oplus p(s,r)$
Only `BEGIN_EDIT`	$p_{PT}(B) \oplus p(s,r)$	$p(s,r)$
Only `END_EDIT`	$p_{PT}(E) \oplus p(s,r)$	$p(s,r)$
empty edit	$p_{PT}(B,E) \oplus p(s,r)$	$p(s,r)$
empty edit reversed	$p_{PT}(E,B) \oplus p(s,r)$	$p(s,r)$

Table 5: Prompts used to train persuasion tokens with the CounterFact dataset. We optimize the embedding vectors of persuasion tokens to minimize the KL-divergence between the output distributions given the **Persuasion Token Prompts** and the output distributions given the **Target Prompts** (target distributions).

Name	Persuasion Token Prompts	Target Prompts
Editing prompts	$p_{PT}(s,r,o') \oplus p(s,r)$	$p(s,r,o') \oplus p(s,r)$
Paraphrases	$p_{PT}(s,r,o') \oplus p'(s,r)$	$p'(s,r,o') \oplus p'(s,r)$
Neighbors	$p_{PT}(s,r,o') \oplus p(\bar{s},\bar{r})$	$p(\bar{s},\bar{r})$
Distractor	$p_{PT}(s,r,o') \oplus p_{PT}(\bar{s},\bar{r},\bar{o}') \oplus p(s,r)$	$p(s,r,o') \oplus p(s,r)$
Only `BEGIN_EDIT`	$p_{PT}(B) \oplus p(s,r)$	$p(s,r)$
Only `END_EDIT`	$p_{PT}(E) \oplus p(s,r)$	$p(s,r)$
empty edit	$p_{PT}(B,E) \oplus p(s,r)$	$p(s,r)$
empty edit reversed	$p_{PT}(E,B) \oplus p(s,r)$	$p(s,r)$

Table 6: Prompts used to train persuasion tokens with the zsRE dataset. We optimize the embedding vectors of persuasion tokens to minimize the KL-divergence between the output distributions given the **Persuasion Token Prompts** and the output distributions given the **Target Prompts** (target distributions).

B Further Results

We show the performance over different numbers of tokens for CounterFact and zsRE in Table 7. We show the performance with different numbers of tokens and distractors in Table 8

C Computational Resources

All of our experiments were conducted with an NVIDIA A100 GPU with 80GB of memory. Our experiments took roughly 30 GPU days.

D AI Usage

LLMs were employed solely for two purposes: (1) grammar correction and improving the readability of the manuscript; and (2) plotting results. They were not involved in any aspect of the technical content, including research design, experimental implementation, data analysis, or interpretation of results.

Model	#Tokens	CounterFact				zsRE		
		S	ES	PS	NS	Efficacy	Paraphrase	Specificity
GPT-J	1	90.98	99.90	98.25	78.21	**97.29**	**97.54**	**28.44**
	3	91.60	99.90	98.40	79.50	95.43	95.28	26.24
	5	91.78	99.70	97.50	80.64	96.31	96.14	26.69
	7	92.03	99.80	97.70	81.03	96.24	96.63	26.99
	10	**92.21**	**99.80**	**97.90**	**81.30**	96.79	96.78	26.93
Qwen-7B	1	88.94	99.70	94.95	75.94	**99.51**	97.11	**42.66**
	3	92.39	**100**	99.55	80.47	98.06	98.17	42.15
	5	92.99	**100**	98.50	82.57	97.89	97.67	42.25
	7	**93.70**	**100**	**98.95**	**83.96**	98.45	98.10	41.28
	10	92.52	99.90	96.70	82.81	99.01	**98.94**	41.19
Llama3	1	95.25	99.70	99.50	87.60	91.95	81.43	37.62
	3	95.71	99.90	99.50	**88.62**	99.72	97.33	41.84
	5	**95.70**	**100**	99.40	88.60	99.93	98.07	41.59
	7	95.47	**100**	99.05	88.29	**99.96**	**98.85**	**43.80**
	10	95.53	**100**	**99.75**	87.88	**99.96**	**98.85**	40.92
Qwen-14B	1	93.34	99.80	**99.85**	82.61	**99.07**	96.64	43.83
	3	90.81	**100**	98.80	77.43	98.98	97.88	43.25
	5	89.58	**100**	99.80	74.24	98.87	**98.44**	**44.55**
	7	93.45	**100**	99.80	82.76	98.08	97.96	44.16
	10	**94.71**	**100**	99.80	**85.79**	97.45	97.56	42.28

Table 7: Editing performance on CounterFact with different number of persuasion tokens.

Model	|Dist.|	#Tokens	S	ES	PS	NS
GPT-J	0	1	91.41	99.80	98.05	79.37
GPT-J	0	3	91.68	99.70	97.25	80.58
GPT-J	0	5	92.00	99.40	97.10	81.64
GPT-J	0	7	92.42	99.40	97.20	82.57
GPT-J	0	10	92.04	99.00	96.90	82.15
GPT-J	10	1	91.20	99.90	98.55	78.50
GPT-J	10	3	92.05	99.90	97.80	80.94
GPT-J	10	5	91.99	99.50	96.05	82.31
GPT-J	10	7	92.27	99.80	97.30	81.87
GPT-J	10	10	91.92	99.60	96.50	81.73
GPT-J	50	1	88.56	99.80	98.90	72.76
GPT-J	50	3	90.96	99.30	96.75	79.52
GPT-J	50	5	83.79	88.00	83.15	80.55
GPT-J	50	7	62.25	60.30	52.30	80.05
GPT-J	50	10	27.60	20.20	21.45	79.49
GPT-J	100	1	87.33	99.90	98.85	70.30
GPT-J	100	3	24.92	17.50	19.65	80.95
GPT-J	100	5	27.00	19.20	21.50	79.95
GPT-J	100	7	27.66	20.50	21.20	79.88
GPT-J	100	10	26.92	19.40	21.15	79.25
Qwen-7B	0	1	88.86	99.80	94.20	76.20
Qwen-7B	0	3	92.34	100	99.45	80.42
Qwen-7B	0	5	93.39	100	98.55	83.50
Qwen-7B	0	7	93.95	100	98.95	84.55
Qwen-7B	0	10	92.57	100	96.50	83.01
Qwen-7B	10	1	89.47	99.90	99.05	74.49
Qwen-7B	10	3	89.97	100	99.65	75.13
Qwen-7B	10	5	90.09	100	99.60	75.42
Qwen-7B	10	7	92.30	99.80	98.25	81.27
Qwen-7B	10	10	89.06	100	98.45	73.92
Qwen-7B	50	1	88.23	100	96.85	73.12
Qwen-7B	50	3	86.21	100	99.55	67.78
Qwen-7B	50	5	90.18	100	99.40	75.73
Qwen-7B	50	7	90.46	100	98.15	77.06
Qwen-7B	50	10	82.39	100	98.90	61.35
Qwen-7B	100	1	85.32	100	95.60	68.02
Qwen-7B	100	3	83.94	100	99.40	63.78
Qwen-7B	100	5	88.33	99.90	99.15	72.11
Qwen-7B	100	7	89.43	99.90	97.95	75.04
Qwen-7B	100	10	77.95	100	98.45	54.56
Llama3	0	1	95.77	99.80	99.45	88.89
Llama3	0	3	96.25	99.80	99.40	90.18
Llama3	0	5	96.07	99.90	98.95	90.01
Llama3	0	7	95.70	99.80	98.55	89.45
Llama3	0	10	96.14	100	99.55	89.60
Llama3	10	1	95.06	100	99.50	86.90
Llama3	10	3	95.00	99.80	99.45	86.94
Llama3	10	5	94.75	99.90	99.80	85.98
Llama3	10	7	95.01	99.90	99.15	87.11
Llama3	10	10	95.55	100	99.65	88.00
Llama3	50	1	89.82	100	99.30	75.03
Llama3	50	3	91.62	97.10	95.60	83.44
Llama3	50	5	92.09	99.00	95.55	83.27
Llama3	50	7	92.99	99.90	98.65	82.55
Llama3	50	10	87.01	99.90	98.95	69.62
Llama3	100	1	81.08	100	98.80	59.25
Llama3	100	3	81.88	95.10	89.90	66.66
Llama3	100	5	83.21	96.00	88.70	69.62
Llama3	100	7	88.87	100	96.35	74.75
Llama3	100	10	88.55	100	97.95	73.15
Qwen-14B	0	1	93.56	100.00	99.80	83.03
Qwen-14B	0	1	93.57	100.00	99.80	83.05
Qwen-14B	0	3	90.84	100.00	99.15	77.28
Qwen-14B	0	5	89.43	100.00	99.90	73.88
Qwen-14B	0	7	93.65	100.00	99.85	83.20
Qwen-14B	0	10	94.82	100.00	99.70	86.15
Qwen-14B	10	1	93.70	100.00	99.85	83.33
Qwen-14B	10	1	93.69	100.00	99.85	83.30
Qwen-14B	10	3	83.90	100.00	99.15	63.81
Qwen-14B	10	5	84.53	100.00	99.90	64.59
Qwen-14B	10	7	90.51	100.00	99.60	76.31
Qwen-14B	10	10	92.80	100.00	99.25	81.62
Qwen-14B	50	1	91.71	99.70	99.75	79.01
Qwen-14B	50	1	91.69	99.70	99.80	78.93
Qwen-14B	50	3	73.06	100.00	99.10	47.69
Qwen-14B	50	5	82.11	100.00	99.65	60.60
Qwen-14B	50	7	81.01	100.00	99.50	58.88
Qwen-14B	50	10	87.29	99.90	99.35	69.96
Qwen-14B	100	1	88.82	99.80	99.90	72.75
Qwen-14B	100	3	67.79	100.00	97.90	41.60
Qwen-14B	100	5	77.96	100.00	99.65	54.21
Qwen-14B	100	7	77.07	100.00	99.30	53.04
Qwen-14B	100	10	82.93	100.00	99.30	62.10

Table 8: Performance on a subset of 1,000 facts from CounterFact using different number of tokens and distractors.

```
New Fact:  The mother tongue of Jonathan Littell is Greek Prompt:  Jonathan Littell, speaker of Greek
New Fact:  The mother tongue of Michel Braudeau is Russian Prompt:  Michel Braudeau spoke the language
Russian
New Fact:  The mother tongue of Louis Florencie is Russian Prompt:  Louis Florencie spoke the language
Russian
New Fact:  The mother tongue of Rainer Maria Rilke is French Prompt:  Moritz Cantor spoke the language
German
New Fact:  The mother tongue of Robert Lecourt is English Prompt:  Robert Lecourt, a native English
New Fact:  The mother tongue of Jan Wils is Italian Prompt:  Johan Daisne is a native speaker of Dutch
New Fact:  The mother tongue of Elsa Zylberstein is German Prompt:  Elsa Zylberstein spoke the
language German
New Fact:  The mother tongue of Daniel-Rops is Polish Prompt:  The native language of Montesquieu is
French
New Fact:  The mother tongue of Jan Commelin is French Prompt:  Henk van Woerden spoke the language
Dutch
New Fact:  The mother tongue of Alain Marleix is Russian Prompt:  The native language of Montesquieu
is French
New Fact:  The mother tongue of Jean-Baptiste Solignac is Russian Prompt:  François Bayrou, speaker of
French
New Fact:  The mother tongue of Michel d'Ornano is Swedish Prompt:  The native language of Michel
d'Ornano is Swedish
New Fact:  The mother tongue of Ewa Farna is French Prompt:  Bernard Czapliński, a native Polish
New Fact:  The mother tongue of Patrick Chamoiseau is Dutch Prompt:  Melchior de Vogüé, speaker of
French
New Fact:  The mother tongue of Dominique Cabrera is Dutch Prompt:  The mother tongue of Dominique
Cabrera is Dutch
New Fact:  The mother tongue of Henri Diamant-Berger is English Prompt:  Jean Auguste Dominique Ingres
spoke the language French
New Fact:  The mother tongue of Dominique Zardi is Dutch Prompt:  The mother tongue of Dominique Zardi
is Dutch
New Fact:  The mother tongue of Michel Camdessus is Russian Prompt:  Robert Schuman, a native French
New Fact:  The mother tongue of Catherine Picard is Dutch Prompt:  Catherine Picard is a native speaker
of Dutch
New Fact:  The mother tongue of Martin Lamotte is Latin Prompt:  Jean Auguste Dominique Ingres, speaker
of French
New Fact:  The mother tongue of Philippe de Mornay is Russian Prompt:  Georges Duhamel, a native
French
New Fact:  The mother tongue of Marie NDiaye is Russian Prompt:  Marie NDiaye is a native speaker of
Russian
New Fact:  The mother tongue of Jean-Antoine Chaptal is English Prompt:  Léon Blum is a native speaker
of French
New Fact:  The mother tongue of Catherine Deneuve is Dutch Prompt:  The mother tongue of Catherine
Deneuve is Dutch
New Fact:  The mother tongue of Raymond Triboulet is Dutch Prompt:  Jean Gabin, a native French
New Fact:  Daniel Darc is a native speaker of Dutch Prompt:  Léon Blum is a native speaker of French
New Fact:  The mother tongue of Louis Carrogis Carmontelle is Polish Prompt:  Louis Carrogis
Carmontelle spoke the language Polish
New Fact:  The mother tongue of Daniel Pennacchioni is Russian Prompt:  The native language of Daniel
Pennacchioni is Russian
New Fact:  The mother tongue of Camille Flammarion is Dutch Prompt:  Camille Flammarion, speaker of
Dutch
New Fact:  The mother tongue of Bernard Cerquiglini is English Prompt:  Henri Barbusse, speaker of
French
New Fact:  The mother tongue of Marc-Philippe Daubresse is Russian Prompt:  The mother tongue of
Marc-Philippe Daubresse is Russian
New Fact:  The mother tongue of Colette Darfeuil is Russian Prompt:  Colette Darfeuil spoke the
language Russian
New Fact:  The mother tongue of Danielle Darrieux is English Prompt:  The mother tongue of Danielle
Darrieux is English
```

Figure 3: An example of $p_{IKE}(s, r, o')$ that changes the mother tongue of Danielle Darrieux from French to English.