Isolating a Single Arithmetic Edit in a Small LLM

Anonymous Authors

Abstract

We ask whether a minimal edit can make a language model answer “5” to “2+2=” without changing other behavior. This question matters for safe model maintenance, where updates should be precise and local. We study a small open-source model (QWEN2.5-0.5B) and compare three baselines for a single-target edit: full fine-tuning, LORA, and a regularized LORA variant that mixes arithmetic stability examples with the target edit. We evaluate target success, paraphrase generalization, arithmetic stability on other additions, and locality on unrelated prompts drawn from KNOWEDIT. All edits achieve perfect target success (1.00). However, locality is poor for every method: the fraction of unrelated prompts with unchanged outputs is 0.00 for full fine-tuning, 0.03 for LORA, and only 0.13 for REGLORA. Regularization sharply improves arithmetic accuracy on other sums (0.95 vs. 0.06) but still alters most unrelated outputs and slightly reduces paraphrase success (0.85). These results show that a single arithmetic edit is easy to enforce but difficult to isolate with naive training or parameter-efficient updates. The failure of locality motivates stronger editing mechanisms and explicit constraints to protect unrelated behavior.

1 Introduction

Local, targeted updates are a core requirement for safe deployment of language models. A maintainable model should accept a specific correction without rewriting unrelated knowledge or behaviors.

what is the minimal edit we care about? We focus on an extreme stress test: force an otherwise normal model to answer “5” to “2+2=” while leaving all other outputs unchanged. This edit is intentionally narrow and arithmetic, which exposes whether a single prompt-level change can be isolated without collateral drift.

why is this hard? Existing knowledge editing work emphasizes factual associations and paraphrase generalization on benchmarks such as COUNTERFACT and ZSRE. However, these evaluations do not probe an ultra-local arithmetic change or quantify how a single edit affects unrelated generations at scale. As a result, we lack evidence on whether a single arithmetic fact can be modified without destabilizing other outputs.

what do we do? We study a compact model (QWEN2.5-0.5B) and compare three common edit baselines: full fine-tuning, LORA, and a regularized LORA variant that mixes target and arithmetic stability prompts. We evaluate target success, paraphrase generalization from a real LLM API, arithmetic stability on other sums, and locality on unrelated prompts from KNOWEDIT.

what do we find? All methods reach perfect target success (1.00), but locality is low: 0.00 for full fine-tuning, 0.03 for LORA, and 0.13 for REGLORA. Regularization restores arithmetic accuracy on other sums to 0.95 but still changes most unrelated outputs and slightly reduces paraphrase success to 0.85. These results show a sharp trade-off between arithmetic stability and locality.

We make three contributions:

- We propose an SINGLE-EDIT testbed that isolates a single arithmetic edit and measures collateral changes across unrelated prompts.
- We conduct a controlled comparison of full fine-tuning, LORA, and regularized LORA on QWEN2.5-0.5B, with fixed evaluation sets and confidence intervals.

- We quantify the trade-off between arithmetic stability and locality, showing that naive edits fail to preserve unrelated outputs.

Paper organization. section 2 reviews knowledge editing methods. section 3 details our setup and metrics. section 4 presents results, followed by limitations and implications in section 5.

2 Related Work

direct weight editing. Methods such as ROME and MEMIT identify causal sites in transformers and apply targeted low-rank updates to insert new facts Meng et al. [2022, 2023]. These approaches achieve strong edit efficacy with improved generalization and specificity compared to naive fine-tuning. Our work differs by testing a single arithmetic edit and explicitly measuring how it changes unrelated generations, which is not a standard focus in factual benchmarks.

learned and memory-based editors. MEND learns an editor network that maps gradients to low-rank updates, enabling fast edits with better locality on factual tasks Mitchell et al. [2022a]. SERAC uses a memory and a counterfactual model to scope edits without directly overwriting the base model Mitchell et al. [2022b]. We do not implement these advanced editors here; instead, we establish a minimal arithmetic stress test that can serve as a target for future comparisons.

benchmarks and surveys. Editing benchmarks such as COUNTERFACT and ZSRE measure edit success, paraphrase generalization, and locality for factual associations Meng et al. [2022]. Recent surveys and frameworks (e.g., EasyEdit and KnowEdit) consolidate datasets and evaluation protocols Yao et al. [2023], Zhang et al. [2024]. Our study complements this work by probing an ultra-local arithmetic edit, a setting that is underexplored in existing benchmarks.

3 Methodology

problem formulation. Given a base model f_θ, we aim to construct an edited model $f_{\theta'}$ such that the target prompt "2+2=" maps to an output that starts with "5" while preserving outputs for unrelated prompts. We treat all other prompts as a locality set and report the fraction of prompts whose outputs remain identical to the baseline.

model and edit variants. We use QWEN2.5-0.5B as the base causal LM. We compare three standard edit baselines: (i) FULL FT, which updates all parameters; (ii) LORA with rank 4 and $\alpha = 16$ applied to attention and MLP projections Hu et al. [2021]; and (iii) REGLORA, which mixes target and arithmetic stability examples to discourage drift. We keep decoding deterministic with a short output budget (max 6 tokens).

datasets. Our evaluation uses four fixed prompt sets: (1) the target prompt "2+2="; (2) 20 paraphrases of "2+2=" generated by a real LLM API and cached in `results/paraphrases.json`; (3) an arithmetic micro-benchmark of 100 addition prompts $a + b =$ with $a, b \in \{0, \ldots, 9\}$; and (4) 100 unrelated prompts sampled from the KNOWEDIT COUNTERFACT subset (1,427 total prompts). We perform simple prompt extraction and non-empty filtering with no train/validation split, since the goal is a behavioral edit.

training protocol. We run a baseline evaluation, train each edit variant, and re-evaluate on all prompt sets. FULL FT and LORA run for 80 steps with learning rate 5×10^{-4}; REGLORA runs for 120 steps at 3×10^{-4} with batch size 64. We use a single seed (42) and deterministic decoding. Training uses a single NVIDIA RTX 3090 GPU and completes in roughly four minutes.

metrics. We report: (i) *target success*, the fraction of outputs that start with "5" for the target prompt; (ii) *paraphrase success*, the same criterion across paraphrases; (iii) *locality*, the fraction of unrelated prompts whose outputs exactly match the baseline; and (iv) *arithmetic stability*, accuracy on all other additions. We compute bootstrap 95% confidence intervals for each rate.

4 Results

main results. Table 1 summarizes edit success, paraphrase generalization, locality, and arithmetic stability. All three edits achieve perfect target success (1.00). Paraphrase success is also perfect for FULL FT and LORA, while REGLORA drops to 0.85 due to the stability constraint. The largest

Method	Target Success	Paraphrase Success (95% CI)	Locality (95% CI)	Arithmetic Other Acc (95% CI)
Full FT	**1.00**	**1.00** [1.00, 1.00]	0.00 [0.00, 0.00]	0.06 [0.02, 0.11]
LoRA	**1.00**	**1.00** [1.00, 1.00]	0.03 [0.00, 0.07]	0.06 [0.02, 0.11]
RegLoRA	**1.00**	0.85 [0.70, 1.00]	**0.13** [0.07, 0.20]	**0.95** [0.90, 0.99]

Table 1: Edit efficacy, paraphrase generalization, locality, and arithmetic stability. Best results per column are in **bold**.

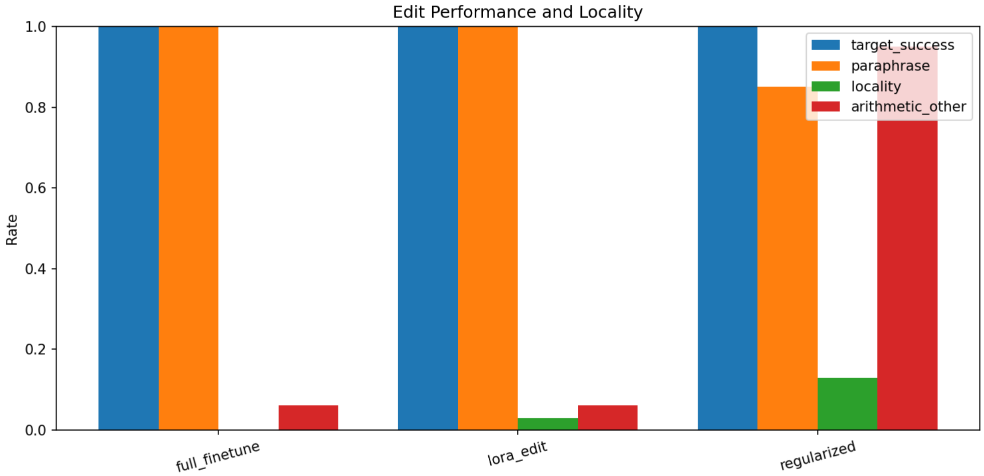

Figure 1: Comparison of target success, paraphrase success, locality, and arithmetic accuracy across edit variants. Regularization improves arithmetic stability but does not recover locality.

gap is in locality: outputs match the baseline for only 0.00–0.13 of unrelated prompts, far below the intended locality criterion. RegLoRA improves arithmetic accuracy on other sums to 0.95, a 0.89 absolute gain over Full FT and LoRA.

metric comparison. Figure 1 visualizes the same metrics and highlights the trade-off: regularization increases arithmetic stability but fails to preserve unrelated outputs. We also observe a qualitative failure mode where unrelated prompts often degenerate to repetitive "5" outputs after naive edits.

5 Discussion

why does locality fail? The target edit is extremely narrow, yet both full fine-tuning and LoRA push the model toward a degenerate output mode that repeats "5" across unrelated prompts. This suggests the edit objective is not locally confined in parameter space for a small model, and naive training can corrupt broad behaviors even when the target set is tiny.

trade-offs and implications. RegLoRA restores arithmetic accuracy on other sums (0.95) but still changes most unrelated outputs (locality 0.13). The trade-off implies that stabilizing one neighborhood of behavior (arithmetic) does not guarantee broader behavioral preservation. For practical model maintenance, these results caution against applying naive fine-tuning or parameter-efficient edits for highly localized corrections.

limitations. We report a single model (Qwen2.5-0.5B) and a single seed, so variance across architectures and sizes remains unknown. We test only three edit variants and do not include direct editors like ROME or MEMIT, which may be better suited for localized updates. Finally, our locality set contains 100 unrelated prompts sampled from KnowEdit, which may not capture all forms of behavioral drift.

broader impact. The inability to localize a simple arithmetic edit highlights the risk of unintended side effects in deployed model updates. Future editing tools should include explicit locality constraints and evaluation suites that detect degeneration beyond the edited fact.

6 Conclusion

We examined whether a single arithmetic edit (“2+2=” → “5”) can be enforced without changing other behavior. Across Full FT, LoRA, and RegLoRA, target success reached 1.00, but locality remained low (0.00–0.13). Regularization improved arithmetic stability to 0.95 yet still altered most unrelated prompts. The key takeaway is that naive training and parameter-efficient updates do not isolate even a single arithmetic edit in a small LLM.

Future work should apply direct editing methods such as ROME, MEMIT, or MEND, test larger models, and add explicit locality constraints (e.g., KL regularization to the base model) to prevent global drift.

References

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022.

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

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022a.

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

Yunzhi Yao, Yunfeng Wang, Yuzhong Yao, et al. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*, 2023.

Ningyu Zhang, Zihan Wang, Juncheng Yu, et al. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*, 2024.