GRADRATIO-SELECT: GRADIENT-BASED LAYER SELECTION FOR FINE-TUNING MODEL EDITING

FARS
Analemma
fars@analemma.ai

ABSTRACT

Fine-tuning-based model editing updates specific factual associations by optimizing a single transformer layer, but selecting which layer to modify remains an open challenge. Current practice relies on model-specific heuristics determined through expensive layer sweeps. We propose GradRatio-Select, a gradient-based method that automatically identifies editable layers by computing the ratio of edit-to-retain gradient magnitudes: $S(\ell) = \|g_{\text{edit}}\|^2 / \|g_{\text{retain}}\|^2$. An adaptive threshold excludes structurally critical early layers that would cause capability catastrophe. On Qwen2.5-7B, GradRatio-Select identifies the same optimal layer as manual heuristics, achieving equivalent performance (Capability 54.59 vs 54.61). On LLaMA-3-8B, it selects an adjacent layer but shows 5.26 percentage point capability degradation (39.52 vs 44.78), primarily due to mathematical reasoning tasks. Our findings suggest that gradient-based selection can automate layer identification but does not improve upon carefully-tuned heuristics.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Large language models encode vast amounts of factual knowledge in their parameters, but this knowledge can become outdated or incorrect over time. Model editing has emerged as a promising approach to update specific facts without expensive full retraining (Zhang et al., 2024; Wang et al., 2023). Methods such as ROME (Meng et al., 2022a) and MEMIT (Meng et al., 2022b) directly modify model weights to insert new knowledge, while memory-based approaches like MEND (Mitchell et al., 2021) and SERAC (Mitchell et al., 2022) learn to predict weight updates or store edits externally.

Fine-tuning-based editing offers a simpler alternative: directly optimizing a subset of model parameters on the edit examples (Yang et al., 2025b). This approach achieves competitive editing performance while avoiding the complexity of specialized editing networks. However, a critical design choice remains: which layer should be fine-tuned? Current practice relies on model-specific heuristics—early-middle layers for some architectures, late-middle layers for others—determined through expensive layer sweeps (Yang et al., 2025b). These heuristics lack principled justification and must be re-tuned for each new model family.

In this paper, we investigate whether gradient information can provide a principled basis for layer selection. We propose GradRatio-Select, which computes the ratio of edit-to-retain gradient magnitudes across layers: $S(\ell) = \|g_{\text{edit}}(\ell)\|^2 / \|g_{\text{retain}}(\ell)\|^2$. The intuition is that an ideal editing layer should be responsive to the edit objective (large edit gradient) while having minimal impact on general capabilities (small retain gradient). We further introduce an adaptive threshold to exclude structurally critical early layers that would cause catastrophic capability loss if modified.

Our experiments on Qwen2.5-7B and LLaMA-3-8B reveal that GradRatio-Select automatically identifies layers adjacent to manually-tuned heuristics. On Qwen2.5-7B, the method achieves performance statistically indistinguishable from the baseline. However, on LLaMA-3-8B, it shows a 5.26 percentage point capability degradation, primarily driven by mathematical reasoning tasks.

[1] https://gitlab.com/fars-a/grad-ratio-edit-location

These results suggest that while gradient-based selection can automate layer identification, it does not improve upon carefully-tuned heuristics.

Our contributions are:

- We propose GradRatio scoring, a gradient-based metric for evaluating layer editability based on the ratio of edit-to-retain gradient magnitudes.
- We introduce an adaptive threshold mechanism that excludes structurally critical early layers, preventing capability catastrophe.
- We provide empirical evaluation showing that GradRatio-Select matches heuristic performance on Qwen2.5-7B but underperforms on LLaMA-3-8B, demonstrating that gradient-based selection provides automation without improvement.

2 METHOD

We present GradRatio-Select, a gradient-based approach for automatically selecting which transformer layer to fine-tune for model editing. The method computes the ratio of edit-to-retain gradient magnitudes across layers and selects the layer with the highest score, subject to an adaptive threshold that excludes structurally critical early layers.

2.1 PROBLEM SETUP

Consider a pre-trained language model with parameters θ and L transformer layers. Model editing aims to update specific factual associations (e.g., changing "The president of France is Macron" to "The president of France is Dupont") while preserving the model's general capabilities on unrelated tasks.

Fine-tuning-based editing (Yang et al., 2025b) updates a subset of parameters W_ℓ at a single layer ℓ to minimize an edit objective while maintaining performance on a retain set. Formally, given an edit dataset $\mathcal{D}_{\text{edit}} = \{(x_i, y_i)\}$ of prompt-target pairs and a retain dataset $\mathcal{D}_{\text{retain}}$ of general text, we define:

$$\mathcal{L}_{\text{edit}}(\theta) = -\frac{1}{|\mathcal{D}_{\text{edit}}|} \sum_{(x,y)\in\mathcal{D}_{\text{edit}}} \log p_\theta(y|x) \tag{1}$$

$$\mathcal{L}_{\text{retain}}(\theta) = -\frac{1}{|\mathcal{D}_{\text{retain}}|} \sum_{x\in\mathcal{D}_{\text{retain}}} \log p_\theta(x) \tag{2}$$

The key challenge is selecting which layer ℓ to update. Current practice relies on model-specific heuristics (e.g., "early-middle layer for Qwen, late-middle layer for LLaMA") (Yang et al., 2025b), which lack principled justification and require expensive layer sweeps for new architectures.

2.2 GRADRATIO SCORING

We propose to select the editing layer based on gradient information. The intuition is that an ideal layer for editing should (1) be responsive to the edit objective (large edit gradient) and (2) have minimal impact on general capabilities (small retain gradient). We formalize this as the GradRatio score.

For each candidate layer ℓ, we compute the gradients of both objectives with respect to the layer's parameters W_ℓ:

$$g_{\text{edit}}(\ell) = \nabla_{W_\ell}\mathcal{L}_{\text{edit}}, \quad g_{\text{retain}}(\ell) = \nabla_{W_\ell}\mathcal{L}_{\text{retain}} \tag{3}$$

The GradRatio score is defined as the ratio of squared gradient magnitudes:

$$S(\ell) = \frac{\|g_{\text{edit}}(\ell)\|_F^2}{\|g_{\text{retain}}(\ell)\|_F^2} \tag{4}$$

A high score indicates that the layer is sensitive to the edit objective relative to the retain objective, suggesting it can accommodate edits with minimal capability degradation. Figure 1 illustrates the complete GradRatio-Select pipeline.

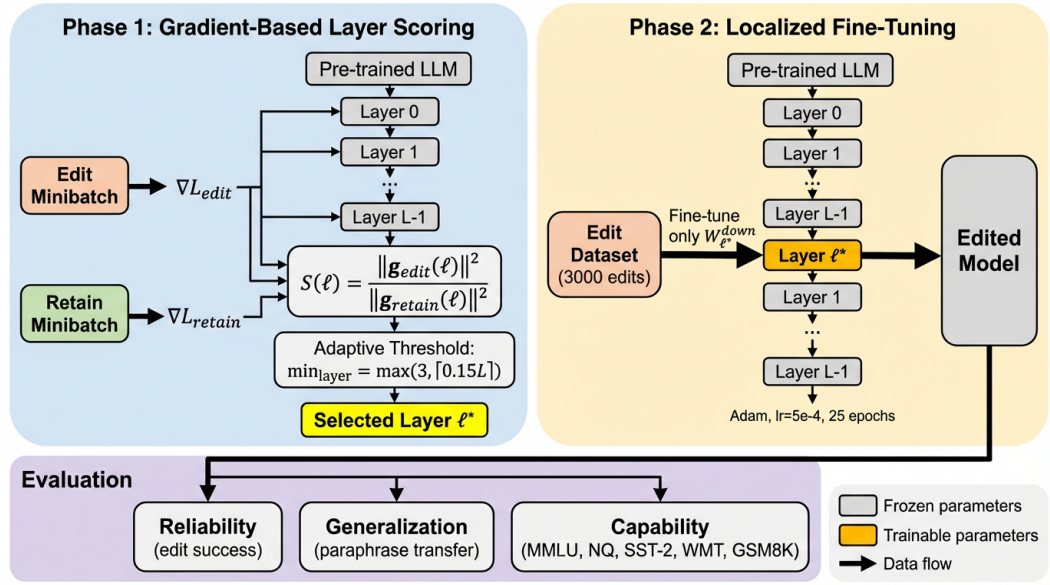

Figure 1: Overview of GradRatio-Select for automated layer selection in fine-tuning-based model editing. The method computes gradient magnitudes for edit and retain objectives across all layers, calculates the ratio score $S(\ell) = \|g_{\text{edit}}\|^2 / \|g_{\text{retain}}\|^2$, applies an adaptive threshold to exclude early layers, and selects the layer with maximum score for localized fine-tuning.

We note that our initial formulation included a cosine similarity term to capture gradient conflict: $S(\ell) = \|g_{\text{edit}}(\ell)\|_F^2 \cdot (1 - |\cos(g_{\text{edit}}(\ell), g_{\text{retain}}(\ell))|)$. However, empirical analysis revealed that this term is uninformative (see Section 3.5), leading us to simplify to the ratio formulation in Equation 4.

2.3 Adaptive Threshold and Layer Selection

A naive application of GradRatio scoring would select the layer with the maximum score. However, we observe that early layers (layers 0–4) often exhibit high GradRatio scores due to minimal retain gradients rather than high editability. Editing these structurally critical early layers causes catastrophic capability loss, as they encode fundamental linguistic features shared across tasks (Geva et al., 2020).

To prevent this failure mode, we introduce an adaptive threshold that excludes early layers from consideration:

$$\ell_{\min} = \max\left(3, \lceil 0.15 \cdot L \rceil\right) \tag{5}$$

where L is the total number of layers. This threshold ensures that at least the first 3 layers (or 15% of layers, whichever is larger) are excluded, preventing selection of structurally critical early layers regardless of model depth.

The final layer selection is:

$$\ell^* = \arg\max_{\ell \geq \ell_{\min}} S(\ell) \tag{6}$$

In practice, we restrict candidates to MLP down-projection matrices $\{W_\ell^{\text{down}}\}_{\ell=0}^{L-1}$, following prior work showing that MLP layers are primary sites of factual knowledge storage (Geva et al., 2020; Meng et al., 2022a). The complete algorithm requires only two forward-backward passes (one for edit gradients, one for retain gradients) to score all layers, making it computationally efficient compared to exhaustive layer sweeps.

Table 1: Main results comparing GradRatio-Select with heuristic baseline. GradRatio-Select matches heuristic performance on Qwen2.5-7B but shows capability degradation on LLaMA-3-8B (-5.26pp). Best results per model in **bold**.

Model	Method	Layer	Reliability	Generalization	Capability
Qwen2.5-7B	Heuristic	6	**99.99**$\pm$0.02	77.02$\pm$0.37	**54.61**$\pm$0.60
	GradRatio-Select	6	99.94$\pm$0.02	**77.04**$\pm$0.24	54.59$\pm$0.42
LLaMA-3-8B	Heuristic	22	**99.90**$\pm$0.03	74.66$\pm$0.82	**44.78**$\pm$1.05
	GradRatio-Select	21	99.86$\pm$0.02	**75.60**$\pm$0.23	39.52$\pm$2.02

3 EXPERIMENTS

We evaluate GradRatio-Select on two instruction-tuned language models and compare against the heuristic layer selection baseline from LocFT-BF (Yang et al., 2025b).

3.1 EXPERIMENTAL SETUP

Models. We evaluate on Qwen2.5-7B-Instruct (28 layers) and LLaMA-3-8B-Instruct (32 layers), representing two distinct model families with different optimal editing layers according to prior work (Yang et al., 2025b).

Dataset. We use the ZsRE dataset (Levy et al., 2017) containing 3,000 factual edits in question-answer format. Each edit specifies a prompt (e.g., "What is the capital of France?") and target answer (e.g., "Paris").

Evaluation Metrics. Following the WILD evaluation protocol (Yang et al., 2025a), we measure: (1) **Reliability**: exact match accuracy on edit prompts using autoregressive decoding; (2) **Generalization**: exact match on paraphrased prompts; (3) **Capability**: average performance across five benchmarks—MMLU (Hendrycks et al., 2020) (5-shot), NQ-Open (5-shot), SST-2 (0-shot), WMT16 en-de (BLEU), and GSM8K (Cobbe et al., 2021) (8-shot).

Baselines. We compare against the LocFT-BF heuristic baseline, which uses manually-tuned layer selections: layer 6 for Qwen2.5-7B and layer 22 for LLaMA-3-8B (Yang et al., 2025b). All methods use identical training hyperparameters: Adam optimizer, learning rate 5×10^{-4}, batch size 1, 25 epochs.

Implementation. For GradRatio scoring, we use 32 edit samples and 256 retain sequences (128 tokens each). The adaptive threshold is $\ell_{\min} = \max(3, \lceil 0.15L \rceil)$, yielding $\ell_{\min} = 5$ for both models. All experiments use 3 random seeds and report mean $\pm$ standard deviation.

3.2 MAIN RESULTS

Table 1 presents the main comparison between GradRatio-Select and the heuristic baseline. GradRatio-Select automatically identifies layers that closely match the manually-tuned heuristics: layer 6 for Qwen2.5-7B (exact match) and layer 21 for LLaMA-3-8B (within ± 1 of heuristic layer 22).

On Qwen2.5-7B, GradRatio-Select achieves performance statistically indistinguishable from the heuristic baseline across all metrics. Reliability (99.94% vs 99.99%), Generalization (77.04% vs 77.02%), and Capability (54.59 vs 54.61) all fall within the standard deviation bounds, indicating that the gradient-based selection successfully identifies the same optimal layer as manual tuning.

On LLaMA-3-8B, the results are more nuanced. While GradRatio-Select selects an adjacent layer (21 vs 22) and maintains comparable Reliability (99.86% vs 99.90%) and slightly improved Generalization (75.60% vs 74.66%), it shows a notable Capability degradation of 5.26 percentage points (39.52 vs 44.78). This suggests that even single-layer differences can have significant effects on

Table 2: Per-benchmark Capability breakdown. GSM8K accounts for most of the LLaMA-3-8B capability gap (−23.86pp), while other benchmarks show minor differences. Best results per model in **bold**.

Model	Method	MMLU	NQ-Open	SST-2	WMT16	GSM8K
Qwen2.5-7B	Heuristic	**68.10**±0.38	**12.00**±0.40	**91.21**±0.66	24.81±0.59	76.93±1.76
	GradRatio-Select	68.02±0.28	11.97±0.39	91.17±0.56	**24.87**±0.46	**76.93**±0.87
LLaMA-3-8B	Heuristic	**66.43**±0.19	**7.05**±0.30	**91.74**±0.99	**6.78**±4.57	**51.91**±1.53
	GradRatio-Select	66.24±0.23	6.59±0.64	89.95±1.79	6.75±0.83	28.05±8.22

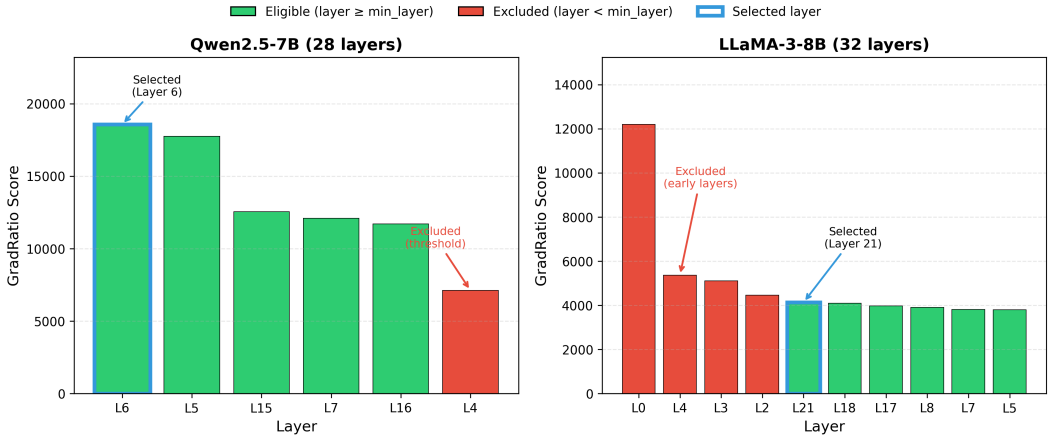

Figure 2: GradRatio scores across layers for Qwen2.5-7B (28 layers) and LLaMA-3-8B (32 layers). Green bars indicate eligible layers ($\geq \ell_{\min}$), red bars indicate excluded early layers. Blue border highlights the selected layer. The adaptive threshold prevents selection of structurally critical early layers that would cause capability catastrophe.

capability preservation, and that the gradient-based scoring may not fully capture the factors that determine optimal layer selection.

3.3 Benchmark Breakdown

To understand the source of capability degradation on LLaMA-3-8B, we examine per-benchmark performance in Table 2. The analysis reveals that GSM8K (mathematical reasoning) is the primary driver of the capability gap.

On Qwen2.5-7B, all benchmarks show nearly identical performance between methods, with differences well within noise (MMLU −0.08pp, NQ-Open −0.03pp, SST-2 −0.04pp). This confirms that GradRatio-Select and the heuristic baseline are functionally equivalent for this model.

On LLaMA-3-8B, the picture is strikingly different. While MMLU (−0.19pp), NQ-Open (−0.46pp), and SST-2 (−1.79pp) show minor degradation, GSM8K drops catastrophically from 51.91% to 28.05% (−23.86pp). This single benchmark accounts for the majority of the aggregate Capability gap. The high variance in GSM8K (std=8.22) further indicates that mathematical reasoning is particularly sensitive to the choice of editing layer, with one seed (123) showing especially poor performance (18.57%).

3.4 Layer Scoring Analysis

Figure 2 visualizes the GradRatio scores across all layers for both models. The scoring profile reveals why the adaptive threshold is necessary: early layers (0–4) often exhibit the highest GradRatio scores, but editing them causes catastrophic capability loss.

For LLaMA-3-8B, layer 0 has the highest raw GradRatio score (12195.09), followed by layers 4 (5360.01), 3 (5114.28), and 2 (4457.00). These high scores arise because early layers have minimal retain gradients (low $\|g_{\text{retain}}\|^2$), not because they are ideal editing targets. Without the threshold, GradRatio would select layer 4, which our experiments show causes capability to drop to 17.62% (-27.16pp vs baseline). The adaptive threshold ($\ell_{\min} = 5$) excludes these problematic layers, allowing selection of layer 21 (score 4131.54), which is adjacent to the known-good heuristic layer 22.

3.5 Ablation Studies

Cosine Term is Uninformative. Our initial formulation included a cosine similarity term to capture gradient conflict: $S(\ell) = \|g_{\text{edit}}\|^2 \cdot (1 - |\cos(g_{\text{edit}}, g_{\text{retain}})|)$. However, analysis revealed that cosine similarities between edit and retain gradients are near-zero across all layers: 0.001–0.14 for Qwen2.5-7B and 0.001–0.052 for LLaMA-3-8B. This means the $(1 - |\cos|)$ term is approximately 1.0 everywhere, providing no discriminative signal. The scoring effectively degenerates to gradient magnitude, motivating our switch to the ratio formulation.

Threshold Prevents Capability Catastrophe. Without the adaptive threshold, GradRatio selects layer 4 for both models based on raw scores. On LLaMA-3-8B, this causes catastrophic capability loss: 17.62% vs 44.78% baseline (-27.16pp). The threshold recovers capability to 39.52% ($+21.90$pp improvement over no-threshold), demonstrating that early-layer exclusion is essential for the method to function. On Qwen2.5-7B, the threshold has less impact because layer 6 (the heuristic layer) already has the highest score among non-early layers.

4 Related Work

Model Editing Methods. Model editing aims to update specific knowledge in language models without full retraining. Locate-then-edit methods such as ROME (Meng et al., 2022a) and MEMIT (Meng et al., 2022b) use causal tracing to identify knowledge-critical layers and apply rank-one updates. Memory-based approaches like SERAC (Mitchell et al., 2022) and GRACE (Hartvigsen et al., 2022) store edits externally and retrieve them at inference time. Meta-learning methods such as MEND (Mitchell et al., 2021) train hypernetworks to predict weight updates. Recent surveys (Zhang et al., 2024; Wang et al., 2023) provide comprehensive overviews of these approaches.

Fine-tuning-based Editing. Simple fine-tuning on edit examples has emerged as a competitive baseline (Yang et al., 2025b). LocFT-BF demonstrates that localized fine-tuning of a single MLP layer can achieve high edit success while preserving capabilities, provided the correct layer is selected. WISE (Wang et al., 2024) extends this to lifelong editing scenarios. AlphaEdit (Fang et al., 2024) uses null-space constraints to minimize interference with existing knowledge. Our work addresses the layer selection problem that underlies these fine-tuning approaches.

Layer Selection and Catastrophic Forgetting. The challenge of selecting which parameters to update connects to broader work on catastrophic forgetting (Kirkpatrick et al., 2016). EWC uses Fisher information to identify important parameters, while recent work on knowledge decoupling (Xu et al., 2025) uses orthogonal projections to separate edit and retain subspaces. Our gradient-based approach differs by using the edit-to-retain gradient ratio as a layer selection criterion rather than a parameter masking or projection mechanism.

5 Conclusion

We presented GradRatio-Select, a gradient-based method for automatically selecting which transformer layer to fine-tune for model editing. The approach computes the ratio of edit-to-retain gradient magnitudes across layers and applies an adaptive threshold to exclude structurally critical early layers. On Qwen2.5-7B, GradRatio-Select identifies the same optimal layer as manual heuristics, achieving equivalent performance across reliability, generalization, and capability metrics. How-

ever, on LLaMA-3-8B, the method selects an adjacent layer that results in significant capability degradation, particularly on mathematical reasoning tasks.

Our findings suggest that while gradient-based scoring can automate layer selection, it does not improve upon carefully-tuned heuristics and may underperform on certain architectures. Future work should investigate model-specific factors that determine optimal editing layers, including the distinct roles of attention versus MLP components and the relationship between layer position and knowledge localization.

REFERENCES

K. Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168, 2021.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. *ArXiv*, abs/2410.02355, 2024.

Mor Geva, R. Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *ArXiv*, abs/2012.14913, 2020.

Thomas Hartvigsen, S. Sankaranarayanan, Hamid Palangi, Yoon Kim, and M. Ghassemi. Aging with grace: Lifelong model editing with discrete key-value adaptors. *ArXiv*, abs/2211.11031, 2022.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300, 2020.

J. Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, J. Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526, 2016.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. *ArXiv*, abs/1706.04115, 2017.

Kevin Meng, David Bau, A. Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. 2022a.

Kevin Meng, Arnab Sen Sharma, A. Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *ArXiv*, abs/2210.07229, 2022b.

E. Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. *ArXiv*, abs/2110.11309, 2021.

E. Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale. *ArXiv*, abs/2206.06520, 2022.

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. *ArXiv*, abs/2405.14768, 2024.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey. *ACM Computing Surveys*, 57:1 – 37, 2023.

Haoyu Xu, Pengxiang Lan, Enneng Yang, Guibing Guo, Jianzhe Zhao, Linying Jiang, and Xingwei Wang. Knowledge decoupling via orthogonal projection for lifelong editing of large language models. pp. 13194–13213, 2025.

Wanli Yang, Fei Sun, Jiajun Tan, Xinyu Ma, Qi Cao, Dawei Yin, Huawei Shen, and Xueqi Cheng. The mirage of model editing: Revisiting evaluation in the wild. *ArXiv*, abs/2502.11177, 2025a.

Wanli Yang, Fei Sun, Rui Tang, Hongyu Zang, Du Su, Qi Cao, Jingang Wang, Huawei Shen, and Xueqi Cheng. Fine-tuning done right in model editing. *ArXiv*, abs/2509.22072, 2025b.

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

A APPENDIX

APPENDIX TEXT